User's Manual

16

The sales of these products are limited for China and Hong Kong.

R7F0C001G/L, R7F0C002G/L

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.2.30 Jun 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 R7F0C001G/L, R7F0C002G/L and design and develop application systems and programs for these devices.
The target products are as follows.

- R7F0C001G2DFB
- R7F0C002G2DFB
- R7F0C001L2DFB
- R7F0C002L2DFB

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

Organization The R7F0C001G/L, R7F0C002G/L manual is separated into two parts: this manual and the software edition (common to the RL78 family).

R7F0C001G/L, R7F0C002G/L User's Manual Hardware	**RL78 Family User's Manual Software**
● Pin functions	● CPU functions
● Internal block functions	● Instruction set
● Interrupts	● Explanation of each instruction
● Other on-chip peripheral functions	
● Electrical specifications	

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

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler.
- To know details of the R7F0C001G/L, R7F0C002G/L 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.
R7F0C001G/L, R7F0C002G/L User’s Manual: Hardware	This manual
RL78 Family User’s Manual: Software	R01US0015E

Documents Related to Flash Memory Programming

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

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 5.5 V which can operate a 1.8 V device at a low voltage
- HALT mode
- STOP mode
- SNOOZE mode

RL78 CPU core

- CISC architecture with 3-stage pipeline
- Minimum instruction execution time: Can be changed from high speed (0.04167 μs: @ 24 MHz operation with high-speed on-chip oscillator) to ultra-low speed (30.5 μs: @ 32.768 kHz operation with subsystem clock)
- Address space: 1 MB
- General-purpose registers: (8-bit register × 8) × 4 banks
- On-chip RAM: 1/1.5 KB

Code flash memory

- Code flash memory: 16/32 KB
- Block size: 1 KB
- Prohibition of block erase and rewriting (security function)
- On-chip debug function
- Self-programming (with flash shield window function)

Data flash memory

- Data flash memory: 2 KB
- Back ground 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 5.5 V

High-speed on-chip oscillator

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

Operating ambient temperature

- T_A = -40 to +85°C

Power management and reset function

- On-chip power-on-reset (POR) circuit
- On-chip voltage detector (LVD) (Select interrupt and reset from 14 levels)

LCD controller/driver (internal voltage boosting method, capacitor split method, and external resistance division method are switchable)

- Number of segment signal output: 39 (35) Note1 to 26 (22)
- Number of common signal output: 4 (8) Note1

Serial interface

- Simplified SPI (CSI Note2): 2 channels
- UART: 1 channel

Timer

- 16-bit timer: 6 channels (remote control output available.)
- 12-bit interval timer: 1 channel
- Real-time clock: 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)

A/D converter

- 8/10-bit resolution A/D converter (V_{DD} = 1.6 to 5.5 V)
- Analog input: 9 to 10 channels
- Internal reference voltage (1.45 V) and temperature sensor Note3

I/O port

- I/O port: 33 to 47 (N-ch open drain I/O [EV_{DD} withstand voltage]: 2)
- Can be set to N-ch open drain, TTL input buffer, and on-chip pull-up resistor
- Different potential interface: Can connect to a 1.8/2.5/3 V device
- On-chip key interrupt function
- On-chip clock output/buzzer output controller

Others

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

Notes **1.** ( ) indicates the number of signal output pins when 8 com is used.

<R> **2.** 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.

<R> **3.** Can be selected only in HS (high-speed main) mode

○ ROM, RAM capacities

Flash ROM	Data flash	RAM	48 pins	64 pins
32 KB	2 KB	1.5 KB Note	R7F0C002G	R7F0C002L
16 KB	2 KB	1 KB Note	R7F0C001G	R7F0C001L

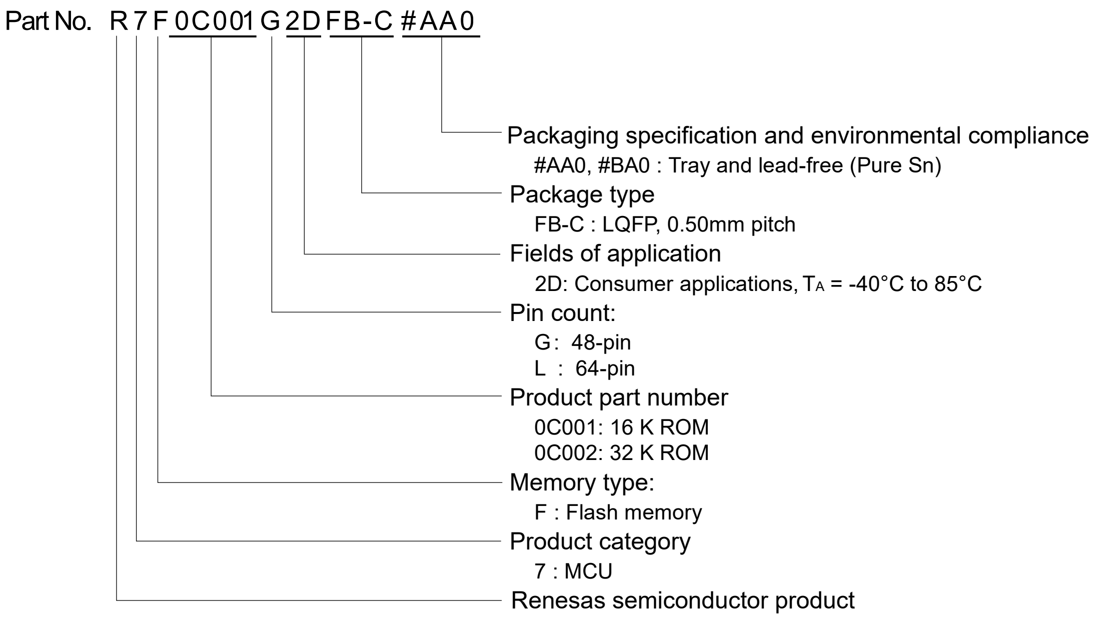
Note In the case of the 1 KB, and 1.5 KB, this is 630 bytes when the self-programming function and data flash function is used. (For details, see **CHAPTER 3 CPU ARCHITECTURE**)

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

1.2 List of Part Numbers

Figure 1-1 Part Number, Memory Size, and Package of R7F0C001G/L, R7F0C002G/L

<R>



<R>

Table 1-1. List of Ordering Part Numbers

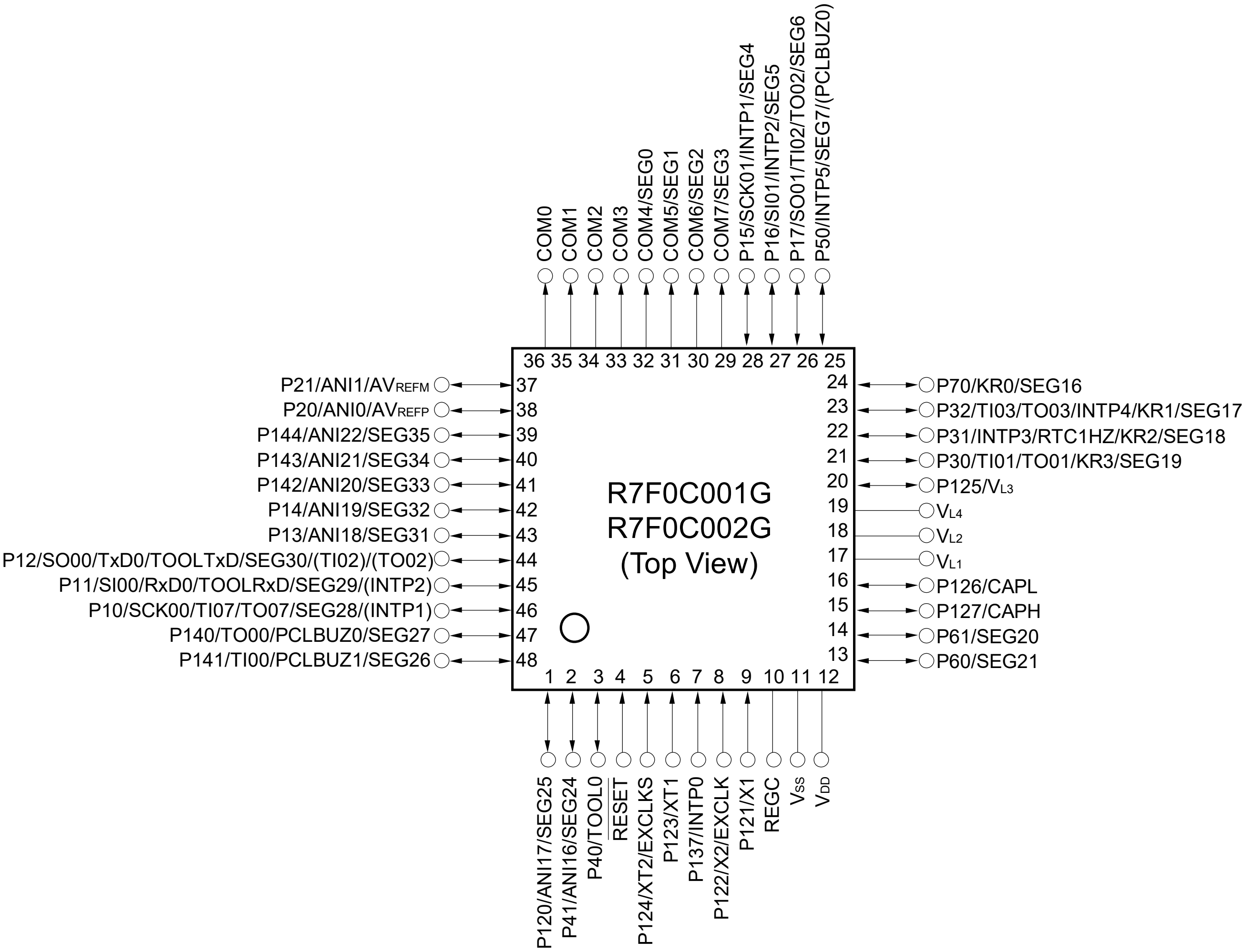
Pin count	Package	Flash ROM	Data flash	Package specification and environmental compliance	Ordering Part Number	
					Product Name	Packaging Specifications
64	64-pin plastic LQFP (fine pitch) (10 x 10)	32KB	2KB	Tray and lead-free (pure Sn)	R7F0C002L2DFB-C	#AA0, #BA0
		16KB			R7F0C001L2DFB-C	
48	48-pin plastic LQFP (fine pitch) (7 x 7)	32KB			R7F0C002G2DFB-C	
		16KB			R7F0C001G2DFB-C	

Caution The R7F0C001G/L, R7F0C002G/L have 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.

1.3 Pin Configuration (Top View)

1.3.1 48-pin products

- 48-pin plastic LQFP (fine pitch) (7 × 7)



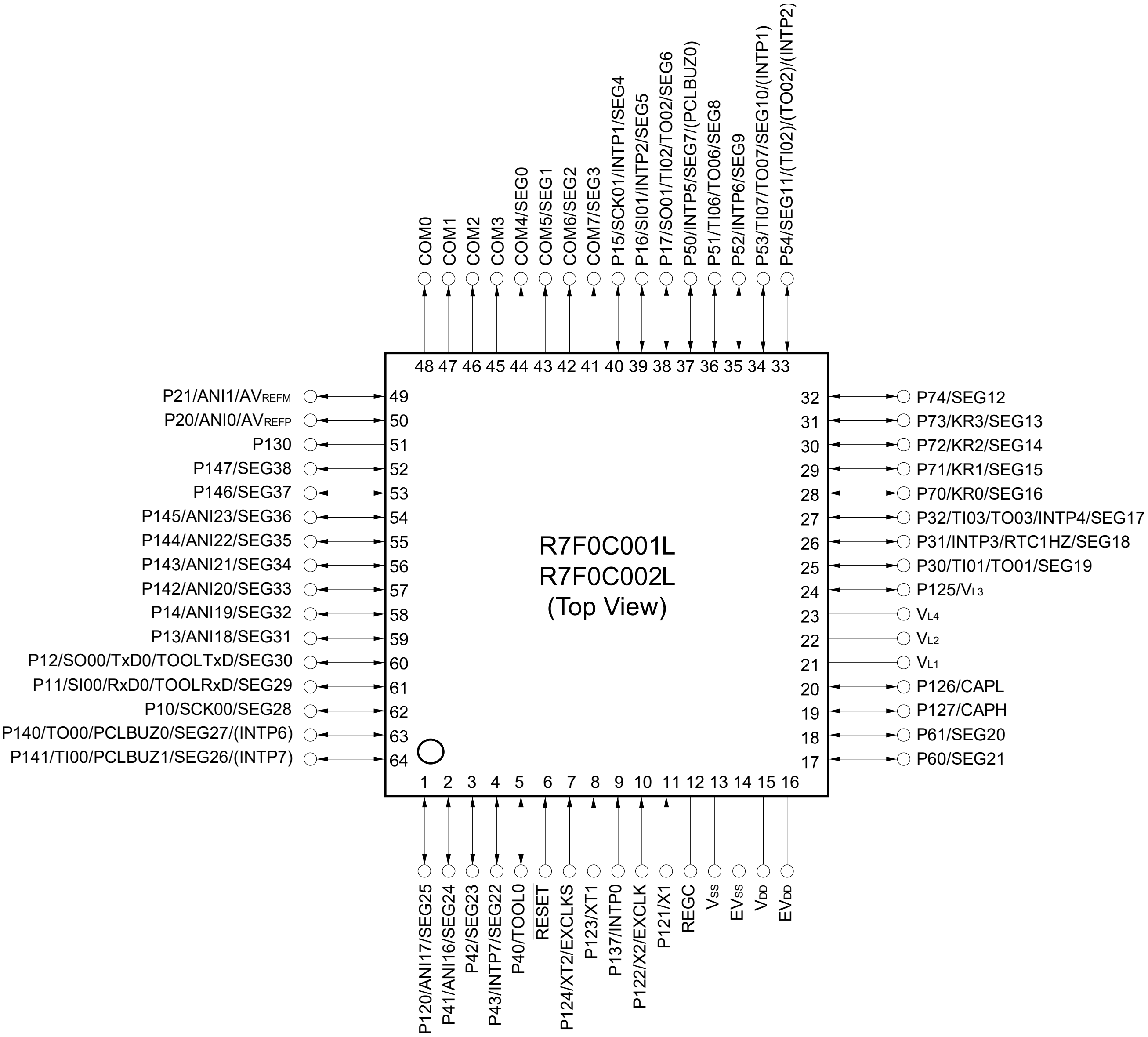
Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

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

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). See **Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR)**.

1.3.2 64-pin products

- 64-pin plastic LQFP (fine pitch) (10 × 10)



Cautions 1. Make EV$_{\mathrm{SS}}$ pin the same potential as V$_{\mathrm{SS}}$ pin.

2. Make EV$_{\mathrm{DD}}$ pin the same potential as V$_{\mathrm{DD}}$ pin.

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

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

2. When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to supply separate powers to the V$_{\mathrm{DD}}$ and EV$_{\mathrm{DD}}$ pins and connect the V$_{\mathrm{SS}}$ and EV$_{\mathrm{SS}}$ pins to separate ground lines.

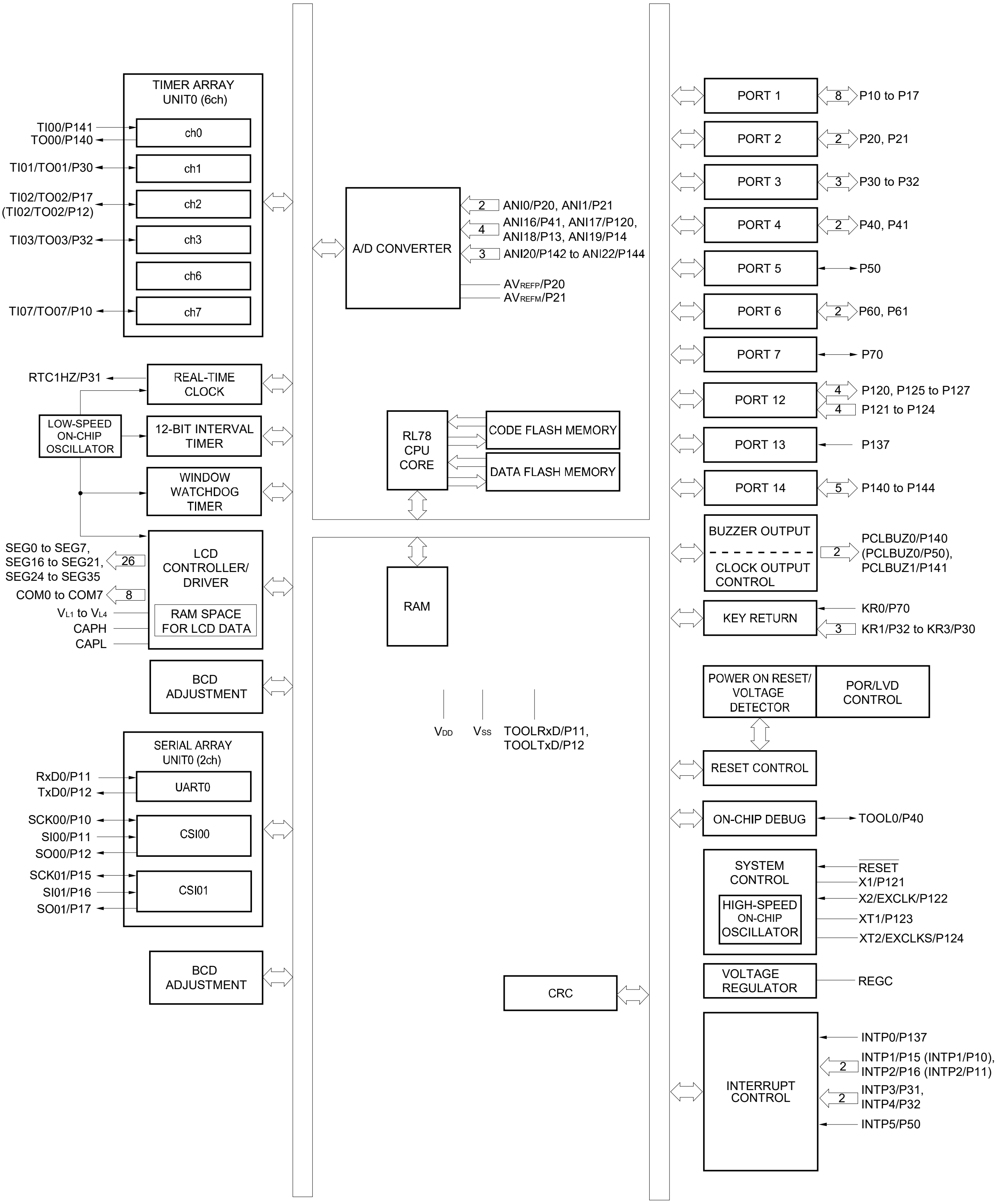
3. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). See **Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR)**.

1.4 Pin Identification

ANI0, ANI1, ANI16 to ANI23:	Analog Input
AV$_{REFM}$:	Analog Reference Voltage Minus
AV$_{REFP}$:	Analog Reference Voltage Plus
CAPH, CAPL:	Capacitor for LCD
COM0 to COM7,	
EV$_{DD}$:	Power Supply for Port
EV$_{SS}$:	Ground for Port
EXCLK:	External Clock Input (Main System Clock)
EXCLKS:	External Clock Input (Subsystem Clock)
INTP0 to INTP7:	Interrupt Request From Peripheral
KR0 to KR3:	Key Return
P10 to P17:	Port 1
P20, P21:	Port 2
P30 to P32:	Port 3
P40 to P43:	Port 4
P50 to P54:	Port 5
P60, P61:	Port 6
P70 to P74:	Port 7
P120 to P127:	Port 12
P130, P137:	Port 13
P140 to P147:	Port 14
PCLBUZ0, PCLBUZ1:	Programmable Clock Output/Buzzer Output
REGC:	Regulator Capacitance
$\overline{\text{RESET}}$:	Reset
RTC1HZ:	Real-time Clock Correction Clock (1 Hz) Output
RxD0:	Receive Data
SCK00, SCK01:	Serial Clock Input/Output
SEG0 to SEG38:	LCD Segment Output
SI00, SI01:	Serial Data Input
SO00, SO01:	Serial Data Output
TI00 to TI03, TI06, TI07:	Timer Input
TO00 to TO03, TO06, TO07:	Timer Output
TOOL0:	Data Input/Output for Tool
TOOLRxD, TOOLTxD:	Data Input/Output for External Device
TxD0:	Transmit Data
V$_{DD}$:	Power Supply
V$_{L1}$ to V$_{L4}$:	LCD Power Supply
V$_{SS}$:	Ground
X1, X2:	Crystal Oscillator (Main System Clock)
XT1, XT2:	Crystal Oscillator (Subsystem Clock)

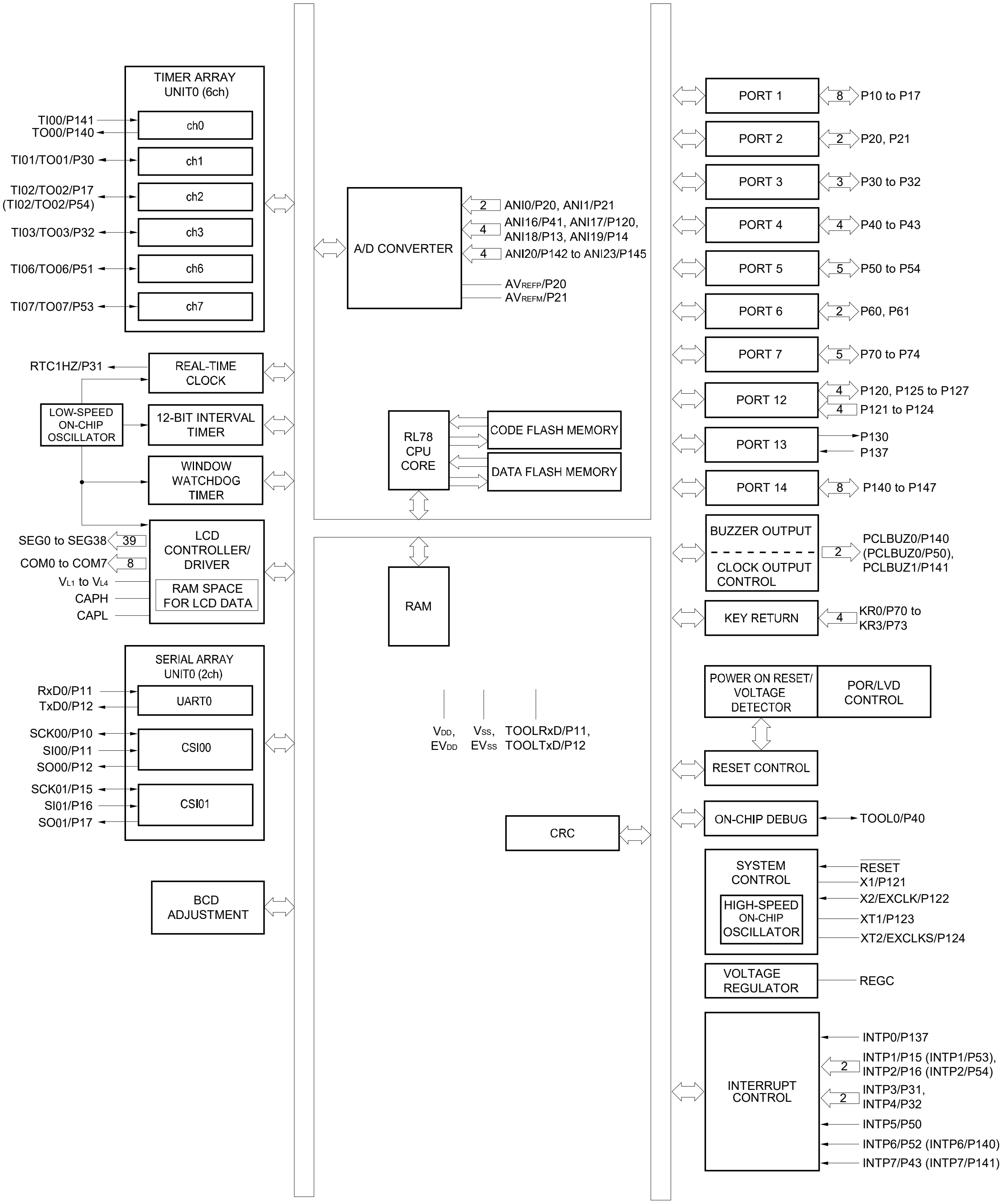
1.5 Block Diagram

1.5.1 48-pin products



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

1.5.2 64-pin products



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

1.6 Outline of Functions

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

(1/2)

Item		48-pin		64-pin	
		R7F0C001G	R7F0C002G	R7F0C001L	R7F0C002L
Code flash memory (KB)		16	32	16	32
Data flash memory (KB)		2	2	2	2
RAM (KB)		1 Note 1	1.5 Note 1	1 Note 1	1.5 Note 1
Memory space		1 MB			
Main system clock	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) HS (high-speed main) operation: 1 to 20 MHz (V_{DD} = 2.7 to 5.5 V), HS (high-speed main) operation: 1 to 16 MHz (V_{DD} = 2.4 to 5.5 V), LS (low-speed main) operation: 1 to 8 MHz (V_{DD} = 1.8 to 5.5 V), LV (low-voltage main) operation: 1 to 4 MHz (V_{DD} = 1.6 to 5.5 V)			
	High-speed on-chip oscillator clock	HS (high-speed main) operation: 1 to 24 MHz (V_{DD} = 2.7 to 5.5 V), HS (high-speed main) operation: 1 to 16 MHz (V_{DD} = 2.4 to 5.5 V), LS (low-speed main) operation: 1 to 8 MHz (V_{DD} = 1.8 to 5.5 V), LV (low-voltage main) operation: 1 to 4 MHz (V_{DD} = 1.6 to 5.5 V)			
Subsystem clock		XT1 (crystal) oscillation , external subsystem clock input (EXCLKS) 32.768 kHz (TYP.): V_{DD} = 1.6 to 5.5 V			
Low-speed on-chip oscillator clock		Internal oscillation 15 kHz (TYP.): V_{DD} = 1.6 to 5.5 V			
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)			
Minimum instruction execution time		0.04167 μs (High-speed on-chip oscillator clock: f_{IH} = 24 MHz operation)			
		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)			
		30.5 μs (Subsystem clock: f_{SUB} = 32.768 kHz operation)			
Instruction set		● Data transfer (8/16 bits) ● Adder and subtractor/logical operation (8/16 bits) ● Multiplication (8 bits × 8 bits) ● Rotate, barrel shift, and bit manipulation (Set, reset, test, and Boolean operation), etc.			
I/O port	Total	33		47	
	CMOS I/O	26		39	
	CMOS input	5		5	
	CMOS output	–		1	
	N-ch open-drain I/O (EV_{DD} tolerance)	2		2	
Timer	16-bit timer	6 channels (with 1 channel remote control output function)			
	Watchdog timer	1 channel			
	Real-time clock (RTC)	1 channel			
	12-bit interval timer (IT)	1 channel			
	Timer output	5 channels (PWM outputs: 4 Note 2)		6 channels (PWM outputs: 5 Note 2)	
	RTC output	1 ● 1 Hz (subsystem clock: f_{SUB} = 32.768 kHz)			

Notes 1. This is 630 bytes when the self-programming function and data flash function is used. (For details, see **CHAPTER 3 CPU ARCHITECTURE**)

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

(2/2)

Item		48-pin		64-pin	
		R7F0C001G	R7F0C002G	R7F0C001L	R7F0C002L
Clock output/buzzer output		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)			
8/10-bit resolution A/D converter		9 channels		10 channels	
Serial interface		● Simplified SPI (CSI): 2 channel/UART: 1 channel			
LCD controller/driver		Internal voltage boosting method, capacitor split method, and external resistance division method are switchable.			
	Segment signal output	26 (22) Note 1		39 (35) Note 1	
	Common signal output	4 (8) Note 1			
Vectored interrupt sources	Internal	16		16	
	External	7		9	
Key interrupt		4			
Reset		● Reset by $\overline{\text{RESET}}$ pin ● Internal reset by watchdog timer ● Internal reset by power-on-reset ● Internal reset by voltage detector ● Internal reset by illegal instruction execution Note 2 ● Internal reset by RAM parity error ● Internal reset by illegal-memory access			
Power-on-reset circuit		● Power-on-reset: 1.51 ±0.04V ● Power-down-reset: 1.50 ±0.04 V			
Voltage detector		● Rising edge : 1.67 V to 4.06 V (14 stages) ● Falling edge : 1.63 V to 3.98 V (14 stages)			
On-chip debug function		Provided			
Power supply voltage		V_{DD} = 1.6 to 5.5 V			
Operating ambient temperature		T_A = –40 to +85 °C			

Notes 1. The values in parentheses are the number of signal outputs when 8 com is used.

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.

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) 48-pin, products

Power Supply	Corresponding Pins
V_{DD}	All pins

(2) 64-pin products

Power Supply	Corresponding Pins
EV_{DD}	● Port pins other than P20, P21, P121 to P124, and P137
V_{DD}	● P20, P21, P121 to P124, and P137 ● $\overline{\text{RESET}}$, REGC pin

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

2.1.1 48-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P10	8-5-7	I/O	Digital input invalid	SCK00/TI07/TO07/SEG28/(INTP1)	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, P11, P15, and P16 can be set to TTL input buffer. Output of P10, P12, P15, and P17 can be set to N-ch open-drain output (V_{DD} tolerance). P13 and P14 can be set to analog input [Note 1].
P11	8-5-1			SI00/RxD0/TOOLRxD/SEG29/(INTP2)	
P12	7-5-7			SO00/TxD0/TOOLTxD/SEG30/(TI02)/(TO02)	
P13	7-10-1		Analog input port	ANI18/SEG31	
P14				ANI19/SEG32	
P15	8-5-7		Digital input invalid	SCK01/INTP1/SEG4	
P16	8-5-1			SI01/INTP2/SEG5	
P17	7-5-7			SO01/TI02/TO02/SEG6	
P20	4-3-1	I/O	Analog input port	ANI0/AV_{REFP}	Port 2. 2-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input [Note 2].
P21				ANI1/AV_{REFM}	
P30	7-5-1	I/O	Digital input invalid	TI01/TO01/KR3/SEG19	Port 3. 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.
P31				INTP3/RTC1HZ/KR2/SEG18	
P32				TI03/TO03/INTP4/KR1/SEG17	
P40	7-1-1	I/O	Input port	TOOL0	Port 4. 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. P41 can be set to analog input [Note 1].
P41	7-10-1		Analog input port	ANI16/SEG24	

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

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

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P50	7-5-1	I/O	Digital input invalid	INTP5/SEG7/ (PCLBUZ0)	Port 5. 1-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.
P60	12-1-4	I/O	Digital input invalid	SEG21	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units.
P61				SEG20	
P70	7-5-1	I/O	Digital input invalid	KR0/SEG16	Port 7. 1-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.
P120	7-10-1	I/O	Analog input port	ANI17/SEG25	Port 12. 4-bit I/O port and 4-bit input port. For P120 and P125 to P127, input/output can be specified in 1-bit units. For P120 and P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port. P120 can be set to analog input $^{\text{Note}}$.
P121	2-2-1	Input	Input port	X1	
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P125	8-5-3	I/O	Digital input invalid	V_{L3}	
P126	8-5-2			CAPL	
P127				CAPH	
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input port.
P140	7-5-1	I/O	Digital input invalid	TO00/PCLBUZ0/ SEG27	Port 14. 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. P142 to p144 can be set to analog input $^{\text{Note}}$.
P141				TI00/PCLBUZ1/ SEG26	
P142	7-10-1		Analog input port	ANI20/SEG33	
P143				ANI21/SEG34	
P144				ANI22/SEG35	

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

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

2.1.2 64-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P10	8-5-7	I/O	Digital input invalid	SCK00/SEG28	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, P11, P15, and P16 can be set to TTL input buffer. Output of P10, P12, P15, and P17 can be set to N-ch open-drain output (EV_{DD} tolerance). P13 and P14 can be set to analog input [Note 1].
P11	8-5-1			SI00/RxD0/ TOOLRxD/SEG29	
P12	7-5-7			SO00/TxD0/ TOOLTxD/SEG30	
P13	7-10-1		Analog input port	ANI18/SEG31	
P14				ANI19/SEG32	
P15	8-5-7		Digital input invalid	SCK01/INTP1/ SEG4	
P16	8-5-1			SI01/INTP2/SEG5	
P17	7-5-7			SO01/TI02/TO02/ SEG6	
P20	4-3-1	I/O	Analog input port	ANI0/AV_{REFP}	Port 2. 2-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input [Note 2].
P21				ANI1/AV_{REFM}	
P30	7-5-1	I/O	Digital input invalid	TI01/TO01/SEG19	Port 3. 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.
P31				INTP3/RTC1HZ/ SEG18	
P32				TI03/TO03/INTP4/ SEG17	
P40	7-1-1	I/O	Input port	TOOL0	Port 4. 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. P41 can be set to analog input [Note 1].
P41	7-10-1		Analog input port	ANI16 /SEG24	
P42	7-5-1		Digital input invalid	SEG23	
P43				INTP7/SEG22	

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

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

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P50	7-5-1	I/O	Digital input invalid	INTP5/SEG7/(PCLBUZ0)	Port 5. 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.
P51				TI06/TO06/SEG8	
P52				INTP6/SEG9	
P53				TI07/TO07/SEG10/(INTP1)	
P54				SEG11/(TI02)/(TO02)/(INTP2)	
P60	12-1-4	I/O	Digital input invalid	SEG21	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units.
P61				SEG20	
P70	7-5-1	I/O	Digital input invalid	KR0/SEG16	Port 7. 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.
P71				KR1/SEG15	
P72				KR2/SEG14	
P73				KR3/SEG13	
P74				SEG12	
P120	7-10-1	I/O	Analog input port	ANI17/SEG25	Port 12. 4-bit I/O port and 4-bit input port. For P120 and P125 to P127, input/output can be specified in 1-bit units. For P120 and P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port. P120 can be set to analog input [Note].
P121	2-2-1	Input	Input port	X1	
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P125	8-5-3	I/O	Digital input invalid	V_{L3}	
P126	8-5-2			CAPL	
P127				CAPH	
P130	1-1-1	Output	Output port	–	Port 13. 1-bit output port and 1-bit input port.
P137	2-1-2	Input	Input port	INTP0	
P140	7-5-1	I/O	Digital input invalid	TO00/PCLBUZ0/SEG27/(INTP6)	Port 14. 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. P142 to P145 can be set to analog input [Note].
P141				TI00/PCLBUZ1/SEG26/(INTP7)	
P142	7-10-1		Analog input port	ANI20/SEG33	
P143				ANI21/SEG34	
P144				ANI22/SEG35	
P145				ANI23/SEG36	
P146	7-5-1		Digital input invalid	SEG37	
P147				SEG38	

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

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

2.2 Functions Other Than Port Pins

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

(1/5)

Function Name	I/O	Function	64-pin	48-pin
ANI0	Input	A/D converter analog input	√	√
ANI1			√	√
ANI16			√	√
ANI17			√	√
ANI18			√	√
ANI19			√	√
ANI20			√	√
ANI21			√	√
ANI22			√	√
ANI23			√	–
CAPH	–	Connecting a capacitor for LCD controller/driver	√	√
CAPL			√	√
COM0	Output	LCD controller/driver common signal outputs	√	√
COM1			√	√
COM2			√	√
COM3			√	√
COM4			√	√
COM5			√	√
COM6			√	√
COM7			√	√
V_{L1}	–	LCD drive voltage	√	√
V_{L2}			√	√
V_{L3}			√	√
V_{L4}			√	√
SEG0	Output	LCD controller/driver segment signal outputs	√	√
SEG1			√	√
SEG2			√	√
SEG3			√	√
SEG4			√	√
SEG5			√	√
SEG6			√	√
SEG7			√	√
SEG8			√	–
SEG9			√	–

Remark √: Mounted

–: Not mounted

(2/5)

Function Name	I/O	Function	64-pin	48-pin
SEG10	Output	LCD controller/driver segment signal outputs	√	–
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			√	–
SEG38			√	–

Remark √: Mounted

–: Not mounted

(3/5)

Function Name	I/O	Function	64-pin	48-pin
INTP0	Input	External interrupt request input	√	√
INTP1			√	√
INTP2			√	√
INTP3			√	√
INTP4			√	√
INTP5			√	√
INTP6			√	–
INTP7			√	–
KR0	Input	Key interrupt input	√	√
KR1			√	√
KR2			√	√
KR3			√	√
PCLBUZ0	Output	Clock output/buzzer output	√	√
PCLBUZ1			√	√
REGC	–	Connecting regulator output stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF: target).	√	√
RTC1HZ	Output	Real-time clock correction clock (1 Hz) output	√	√
$\overline{\text{RESET}}$	Input	System reset input	√	√
RxD0	Input	Serial data input to UART0	√	√
SCK00	I/O	Clock input/output for CSI00 and CSI01	√	√
SCK01			√	√
SI00	Input	Serial data input to CSI00 and CSI01	√	√
SI01			√	√
SO00	Output	Serial data output from CSI00 and CSI01	√	√
SO01			√	√

Remark √: Mounted
–: Not mounted

(4/5)

Function Name	I/O	Function	64-pin	48-pin
TI00	Input	External count clock input to 16-bit timer 00	√	√
TI01		External count clock input to 16-bit timer 01	√	√
TI02		External count clock input to 16-bit timer 02	√	√
TI03		External count clock input to 16-bit timer 03	√	√
TI06		External count clock input to 16-bit timer 06	√	–
TI07		External count clock input to 16-bit timer 07	√	√
TO00	Output	16-bit timer 00 output	√	√
TO01		16-bit timer 01 output	√	√
TO02		16-bit timer 02 output	√	√
TO03		16-bit timer 03 output	√	√
TO06		16-bit timer 06 output	√	–
TO07		16-bit timer 07 output	√	√
TxD0	Output	Serial data output from UART0	√	√
X1	–	Resonator connection for main system clock	√	√
X2	–		√	√
EXCLK	Input	External clock input for main system clock	√	√
EXCLKS	Input	External clock input for subsystem clock	√	√
XT1	–	Resonator connection for subsystem clock	√	√
XT2	–		√	√

Remark √: Mounted
–: Not mounted

(5/5)

Function Name	I/O	Function	64-pin	48-pin
V_{DD}	–	<48-pin > Positive power supply for all pins <64-pin > Positive power supply for P20, P21, P121 to P124, P137 and $\overline{\text{RESET}}$ pin	√	√
EV_{DD}	–	<64-pin> Positive power supply for ports (other than P20, P21, P121 to P124, P137) and pins other ports (except for the $\overline{\text{RESET}}$ pin)	√	–
AV_{REFP}	Input	A/D converter reference potential (+ side) input	√	√
AV_{REFM}	Input	A/D converter reference potential (– side) input	√	√
V_{SS}	–	<48-pin > Ground potential for all pins <64-pin> Ground potential for P20, P21, P121 to P124, P137 and $\overline{\text{RESET}}$ pin	√	√
EV_{SS}	–	<64-pin > Ground potential for ports (other than P20, P21, P121 to P124, P137) and pins other ports (except for the $\overline{\text{RESET}}$ pin)	√	–
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	√	√

Remark √: Mounted
–: Not mounted

2.2.2 Description of Functions

Function Name	I/O	Function
COM0 to COM7	Output	LCD controller/driver common signal outputs
SEG0 to SEG38	Output	LCD controller/driver segment signal outputs
V_{L1}, V_{L2}, V_{L3}, V_{L4}	–	LCD drive voltage
CAPH, CAPL	–	Connecting a capacitor for LCD controller/driver
ANI0, ANI1, ANI16 to ANI23	Input	A/D converter analog input (see **Figure 11-39 Analog Input Pin Connection**)
INTP0 to INTP7	Input	External interrupt request input Specified the valid edge: Rising edge, falling edge, or both rising and falling edges
KR0 to KR3	Input	Key interrupt input
PCLBUZ0, PCLBUZ1	Output	Clock output/buzzer output
REGC	–	Pin for connecting regulator output stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.
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	Input	Serial data input pin of serial interface UART0
TxD0	Output	Serial data output pin of serial interface UART0
SCK00, SCK01	I/O	Serial clock I/O pins of serial interfaces CSI00 and CSI01
SI00, SI01	Input	Serial data input pins of serial interfaces CSI00 and CSI01
SO00, SO01	Output	Serial data output pins of serial interfaces CSI00 and CSI01
TI00 to TI03, TI06, TI07	Input	The pins for inputting an external count clock/capture trigger to 16-bit timers 00 to03, 06, 07.
TO00 to TO03, TO06, TO07	Output	Timer output pins of 16-bit timers 00 to 03, 06, 07
X1, X2	–	Resonator connection for main system clock
EXCLK	Input	External clock input for main system clock
XT1, XT2	–	Resonator connection for subsystem clock
EXCLKS	Input	External clock input for subsystem clock

Function Name	I/O	Function
V_{DD}	–	<48-pin > Positive power supply for all pins <64-pin > Positive power supply for P20, P21, P121 to P124, P137 and other than ports
EV_{DD}	–	<64-pin> Positive power supply for ports (other than P20, P21, P121 to P124, P137)
AV_{REFP}	Input	A/D converter reference voltage (+ side) input
AV_{REFM}	Input	A/D converter reference voltage (– side) input
V_{SS}	–	<48-pin, > Ground potential for all pins <64-pin> Ground potential for P20, P21, P121 to P124, P137 and other than ports
EV_{SS}	–	<64-pin> Ground potential for ports (other than P20, P21, P121 to P124, P137)
TOOLRxD	Input	UART reception pin for the external device connection used during flash memory programming
TOOLTxD	Output	UART transmission pin for the external device connection used during flash memory programming
TOOL0	I/O	Data I/O for flash memory programmer/debugger

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

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

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

For details, see **23.4 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} and EV_{DD} to EV_{SS} lines.

2.3 Connection of Unused Pins

Table 2-3 shows the types of pin I/O circuits and the recommended connections of unused pins.

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

Table 2-3. Connection of Unused Pins (64-pin products) (1/3)

Pin Name	I/O	Recommended Connection of Unused Pins
P10/SCK00/SEG28	I/O	<When setting to port I/O> Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P11/SI00/RxD0/TOOLRxD/SEG29		
P12/SO00/TxD0/TOOLTxD/SEG30		
P13/ANI18/SEG31		
P14/ANI19/SEG32		
P15/SCK01/INTP1/SEG4		
P16/SI01/INTP2/SEG5		
P17/SO01/TI02/TO02/SEG6		
P20/ANI0/AV_{REFP}		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P21/ANI1/AV_{REFM}		
P30/TI01/TO01/SEG19		<When setting to port I/O> Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P31/INTP3/RTC1HZ/SEG18		
P32/TI03/TO03/INTP4/SEG17		
P40/TOOL0		Input: Independently connect to EV_{DD} via a resistor or leave open. Output: Leave open.
P41/ANI16/ SEG24		<When setting to port I/O> Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P42/SEG23		
P43/INTP7/SEG22		

Remark With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

Table 2-3. Connection of Unused Pins (64-pin products) (2/3)

Pin Name	I/O	Recommended Connection of Unused Pins
P50/INTP5/SEG7/(PCLBUZ0)	I/O	<When setting to port I/O> Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P51/TI06/TO06/SEG8		
P52/INTP6/SEG9		
P53/TI07/TO07/SEG10/(INTP1)		
P54/SEG11/(TI02)/(TO02)/(INTP2)		
P60/SEG21		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Set the port's output latch to 0 and leave the pins open, or set the port's output latch to 1 and independently connect the pins to EV_{DD} or EV_{SS} via a resistor.
P61/SEG20		
P70/KR0/SEG16		<When setting to port I/O> Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P71/KR1/SEG15		
P72/KR2/SEG14		
P73/KR3/SEG13		
P74/SEG12		
P120/ANI17/SEG25		
P121/X1	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK		
P123/XT1		
P124/XT2/EXCLKS		
P125/V_{L3}	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P126/CAPL		
P127/CAPH		
P130	Output	Leave open.
P137/INTP0	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P140/TO00/PCLBUZ0/SEG27/(INTP6)	I/O	<When setting to port I/O> Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open. <When setting to segment output> Leave open.
P141/TI00/PCLBUZ1/SEG26/(INTP7)		
P142/ANI20/SEG33		
P143/ANI21/SEG34		
P144/ANI22/SEG35		
P145/ANI23/SEG36		
P146/SEG37		
P147/SEG38		
$\overline{\text{RESET}}$	Input	Connect directly or via a resistor to V_{DD}.
REGC	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).

Remarks 1. With products not provided with an EV_{DD} or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

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

Table 2-3. Connection of Unused Pins (64-pin products) (3/3)

Pin Name	I/O	Recommended Connection of Unused Pins
COM0 to COM3	Output	Leave open.
COM4/SEG0		
COM5/SEG1		
COM6/SEG2		
COM7/SEG3		
V_{L1}, V_{L2}, V_{L4}	–	

2.4 Block Diagrams of Pins

Figures 2-1 to 2-14 show the block diagrams of the pins described in **2.1.1 48-pin products** to **2.1.2 64-pin products**.

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

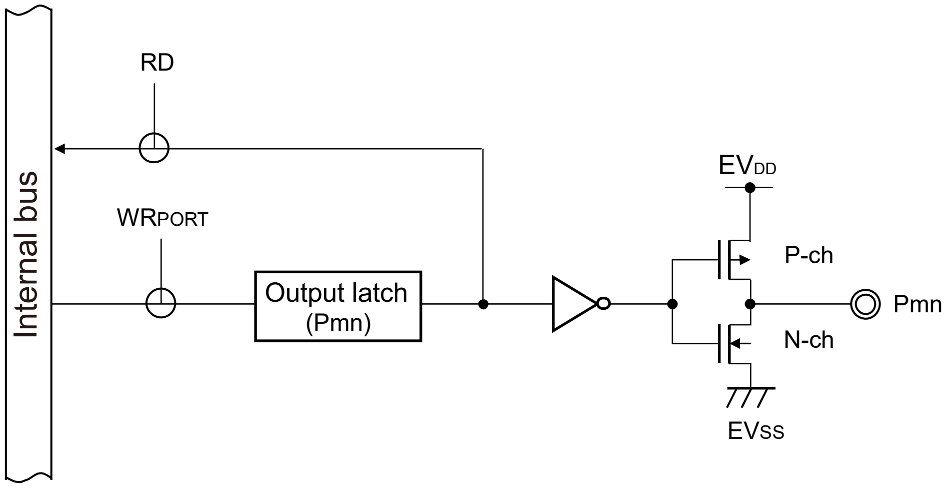


Figure 2-2. Pin Block Diagram for Pin Type 2-1-1

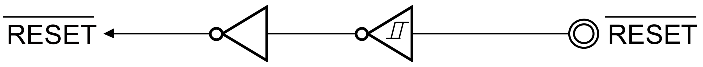
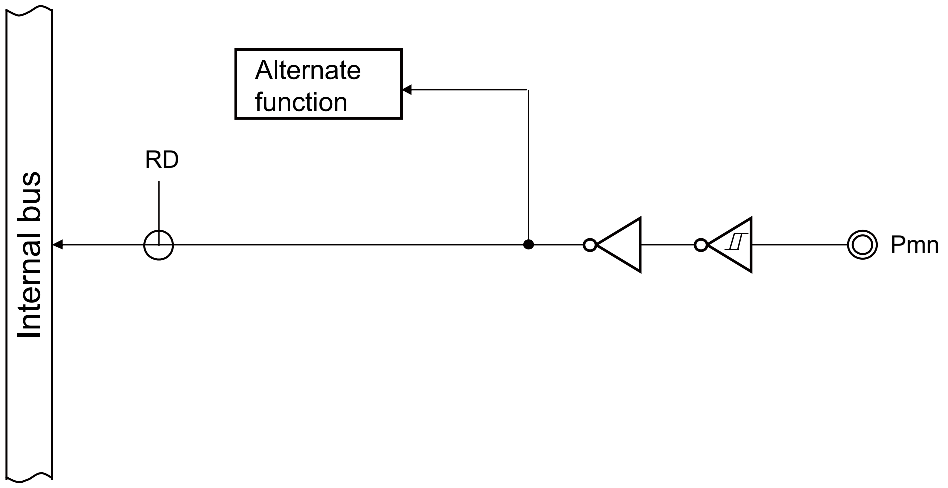
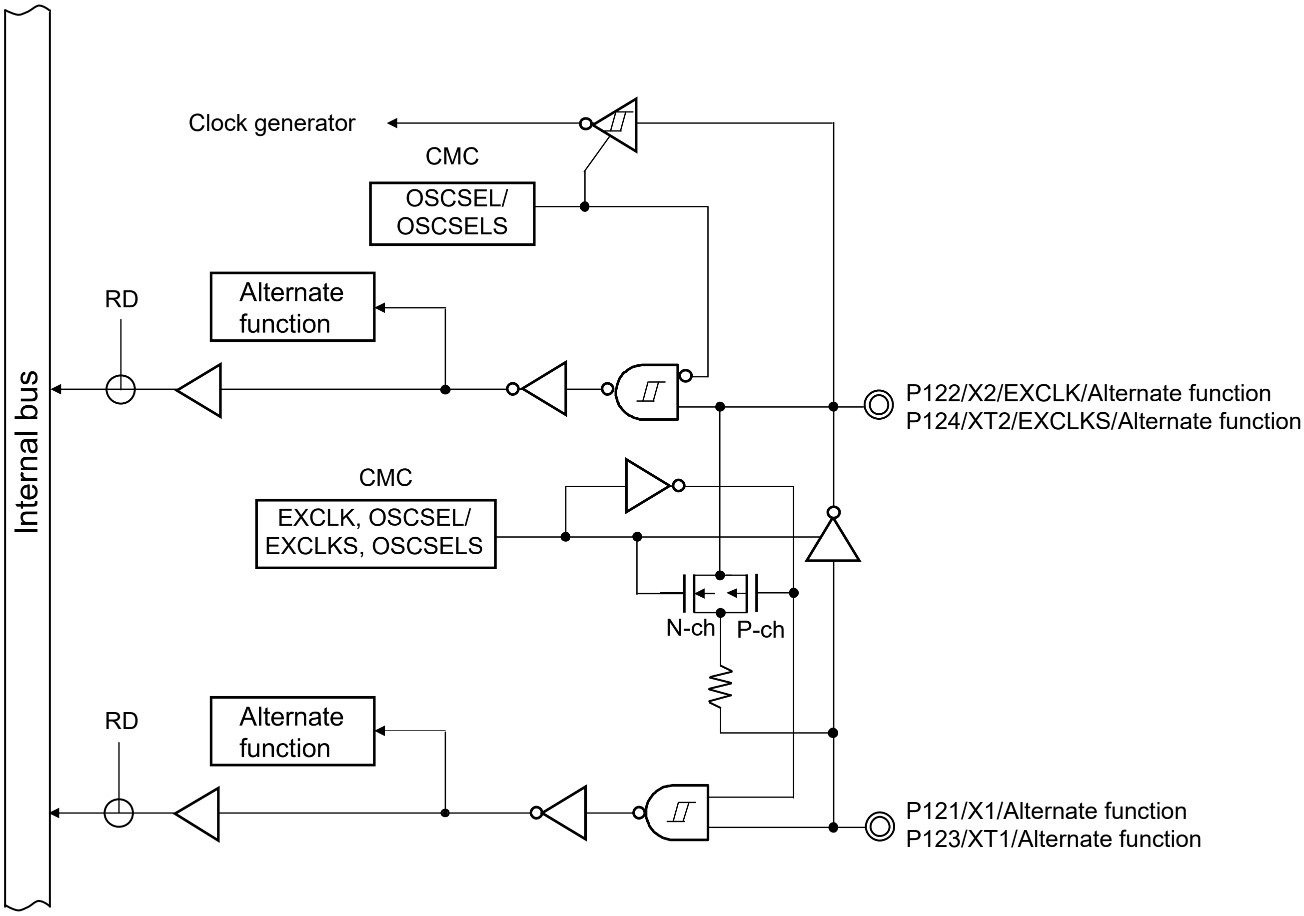


Figure 2-3. Pin Block Diagram for Pin Type 2-1-2



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

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



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

Figure 2-5. Pin Block Diagram for Pin Type 4-3-1

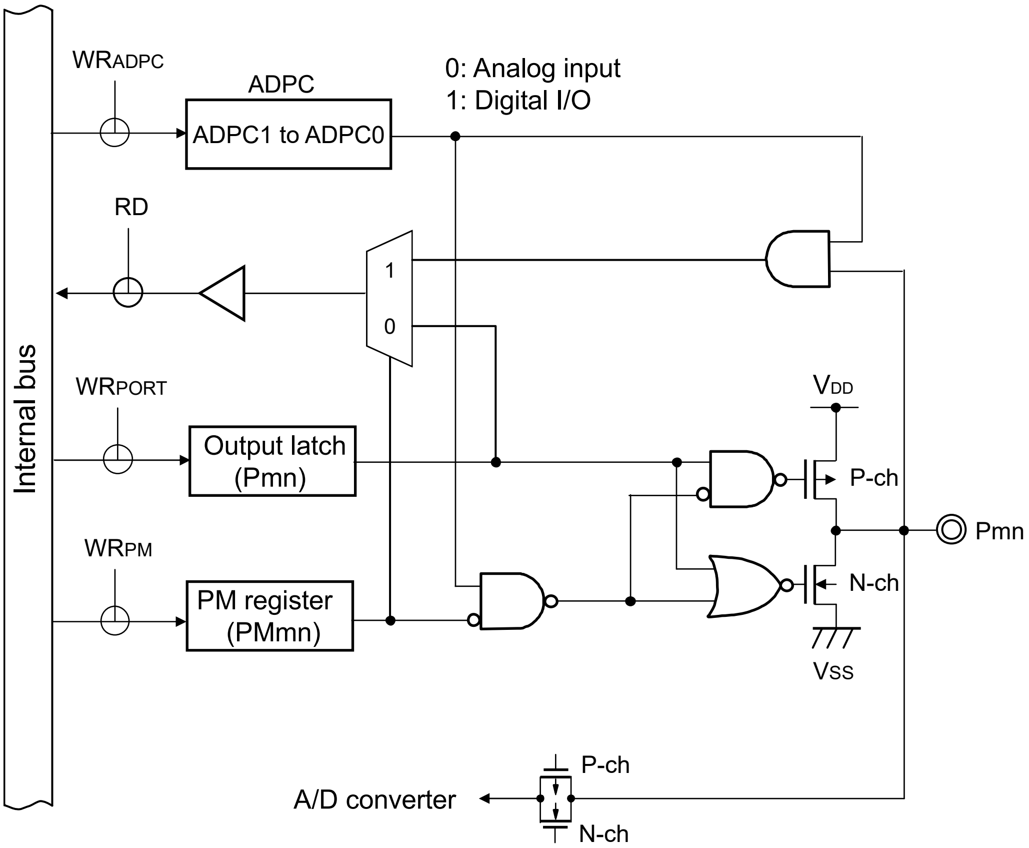
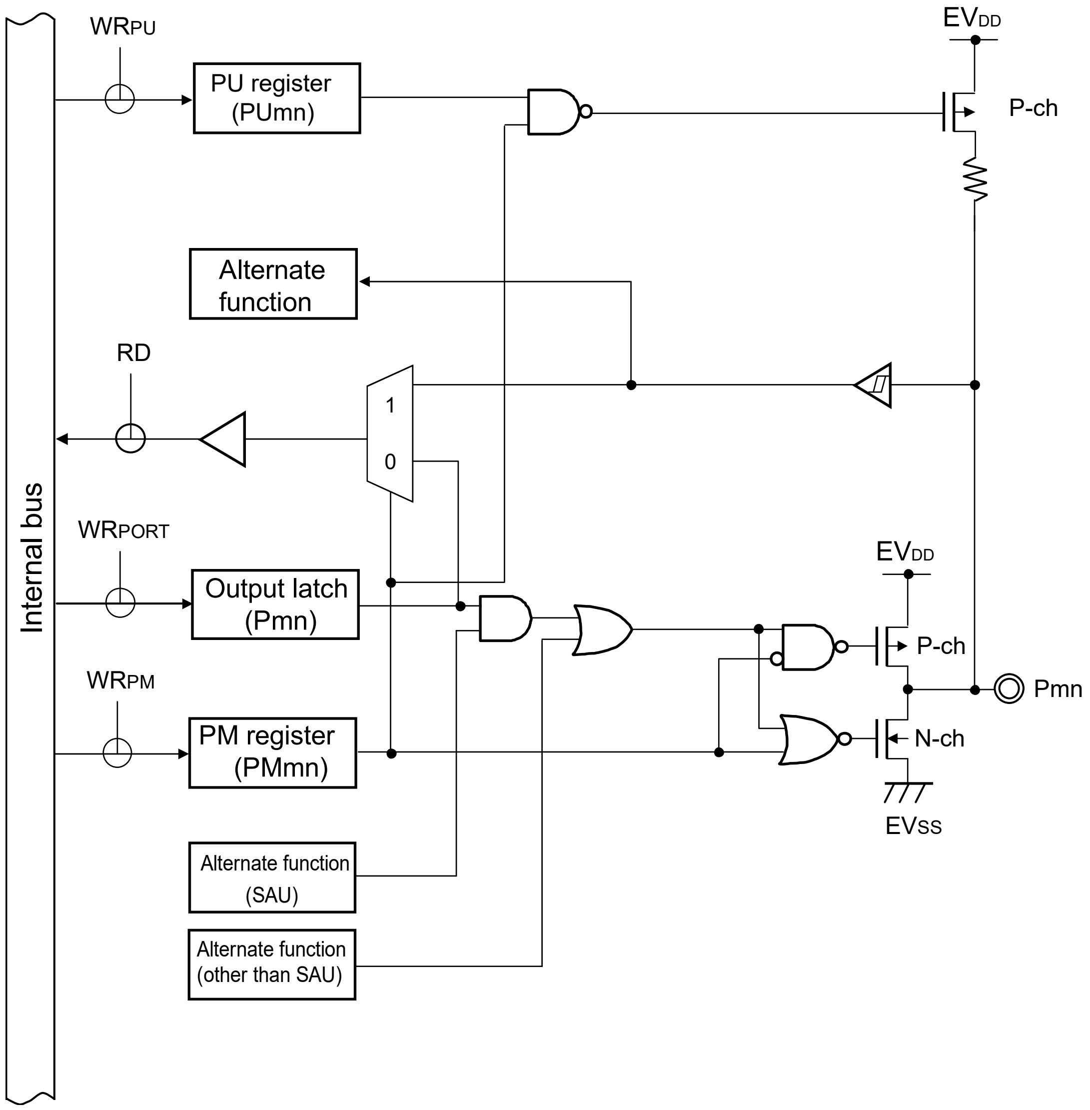


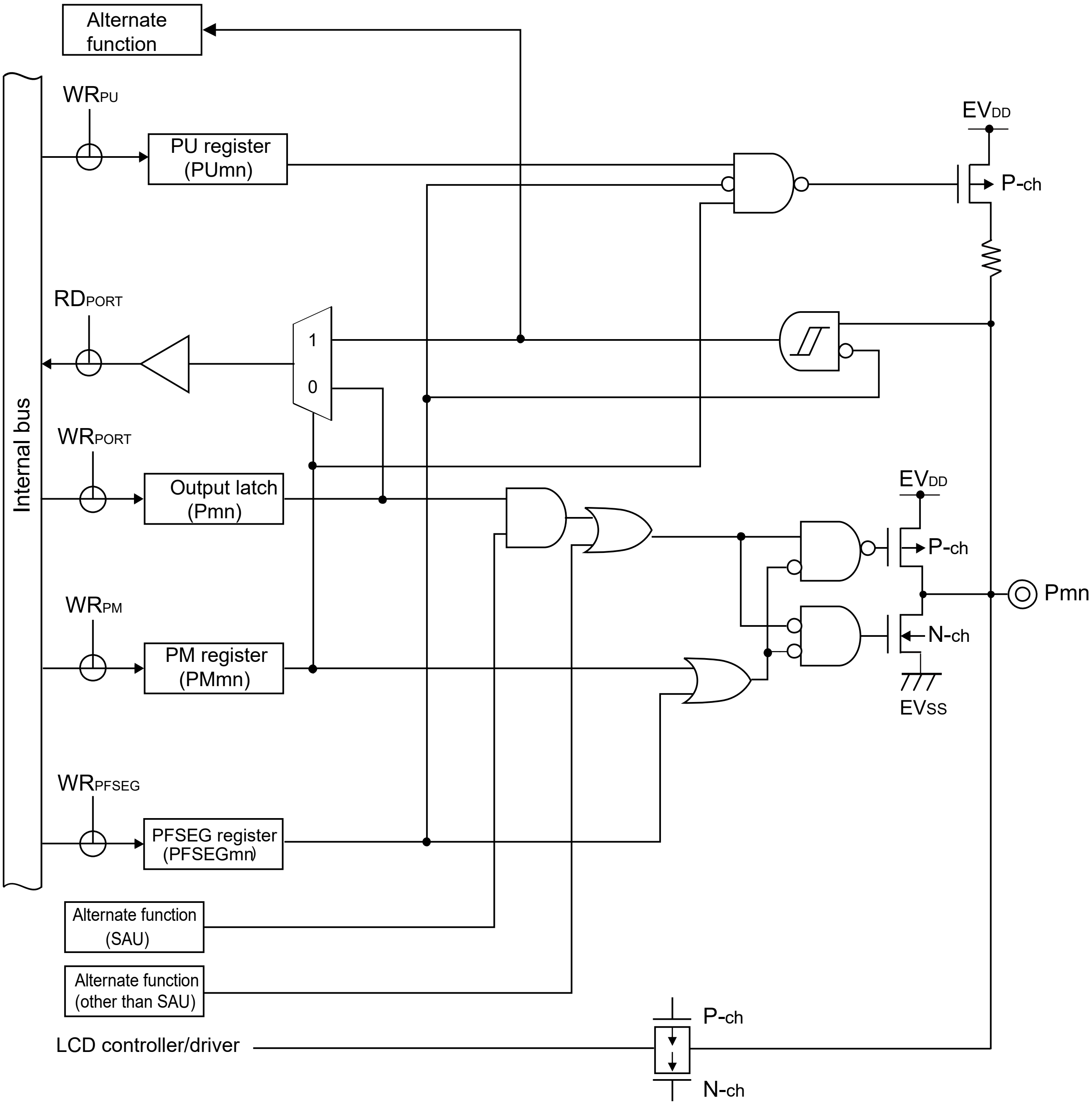
Figure 2-6. Pin Block Diagram for Pin Type 7-1-1



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

2. SAU: Serial array unit

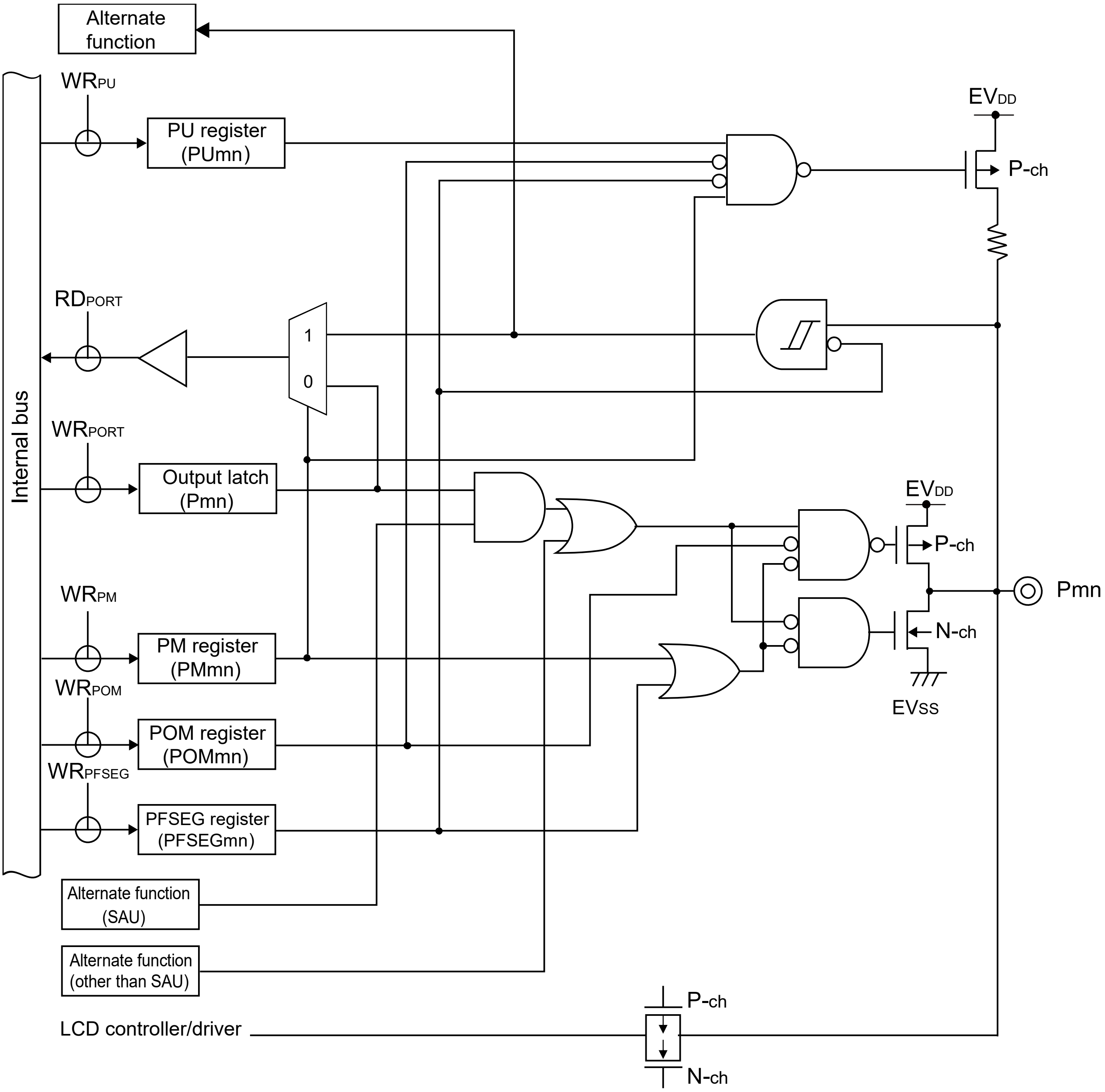
Figure 2-7. Pin Block Diagram for Pin Type 7-5-1



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

2. SAU: Serial array unit

Figure 2-8. Pin Block Diagram for Pin Type 7-5-7

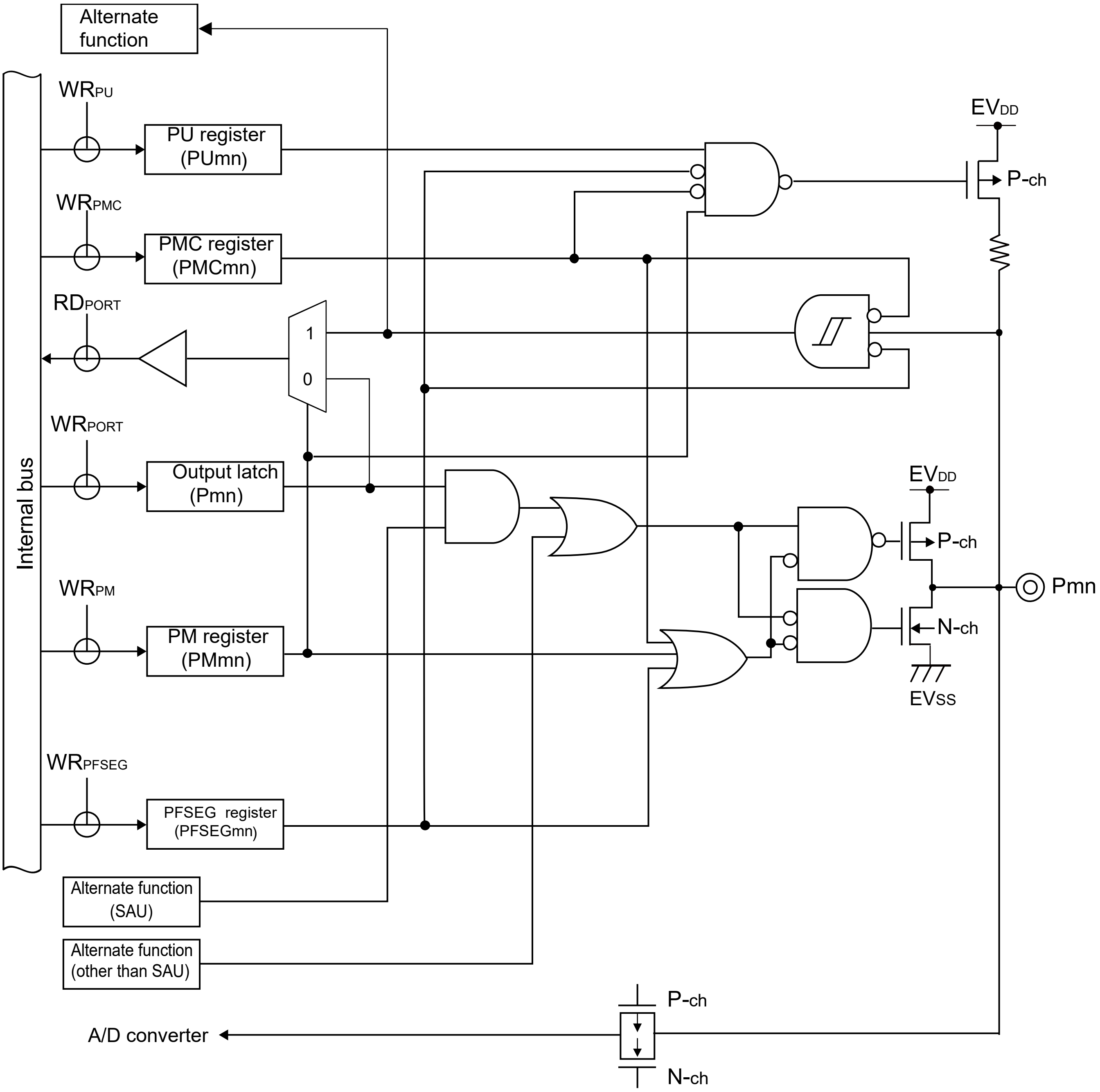


Caution A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

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

2. SAU: Serial array unit

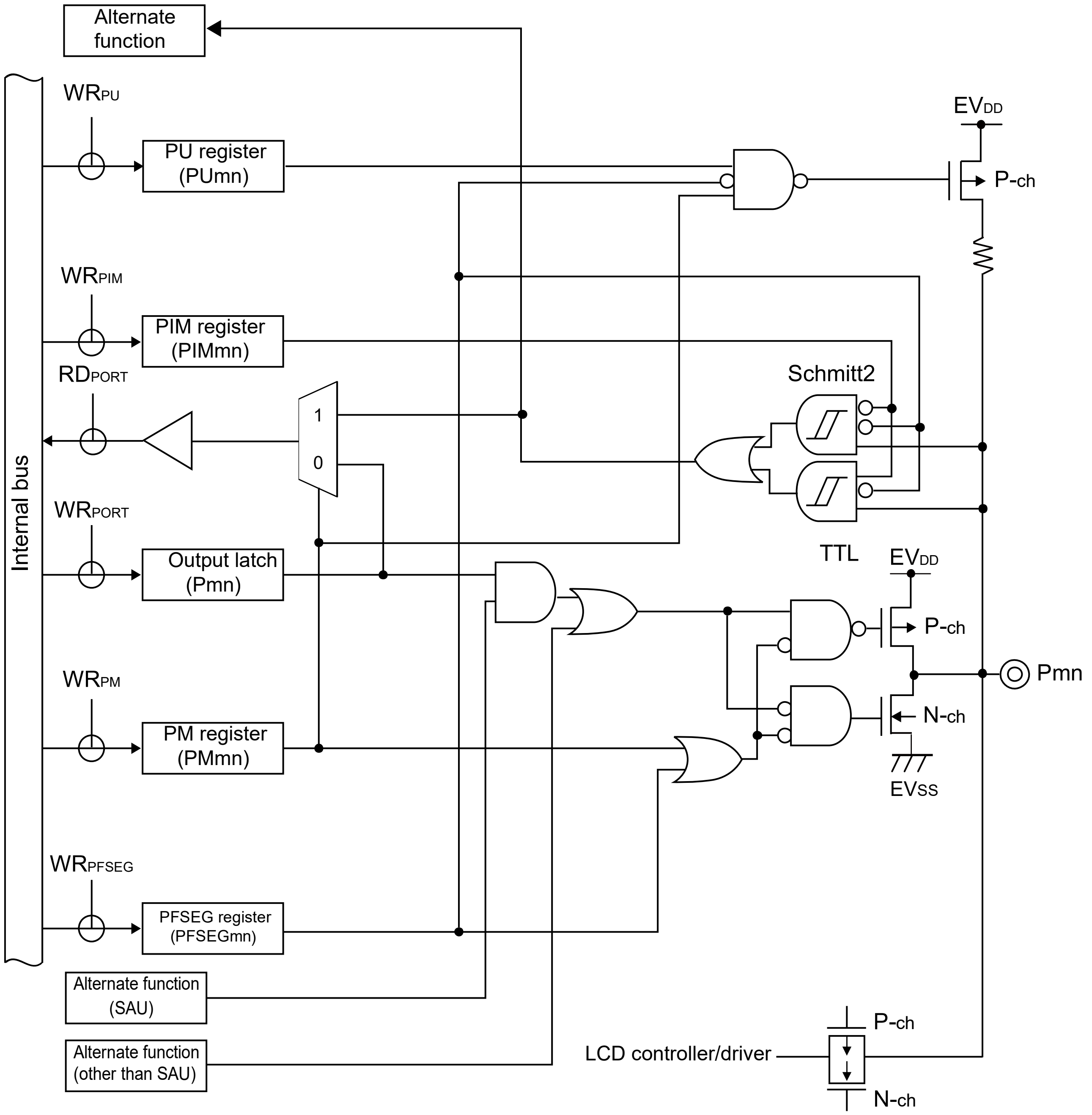
Figure 2-9. Pin Block Diagram for Pin Type 7-10-1



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

2. SAU: Serial array unit

Figure 2-10. Pin Block Diagram for Pin Type 8-5-1

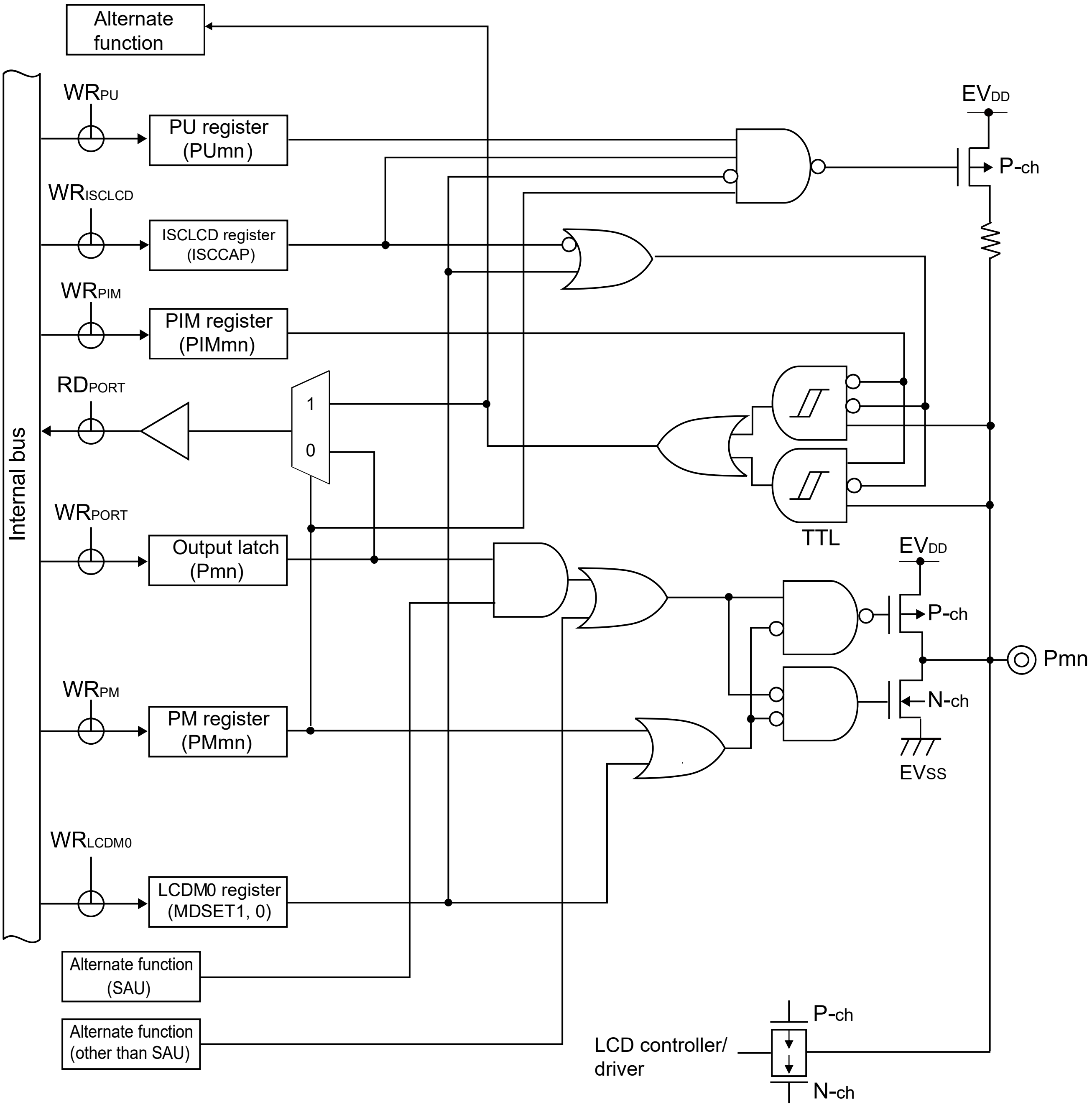


Caution Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, a through current may flow through in the case of high level input. It is recommended to input a low level to prevent a through current.

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

2. SAU: Serial array unit

Figure 2-11. Pin Block Diagram for Pin Type 8-5-2

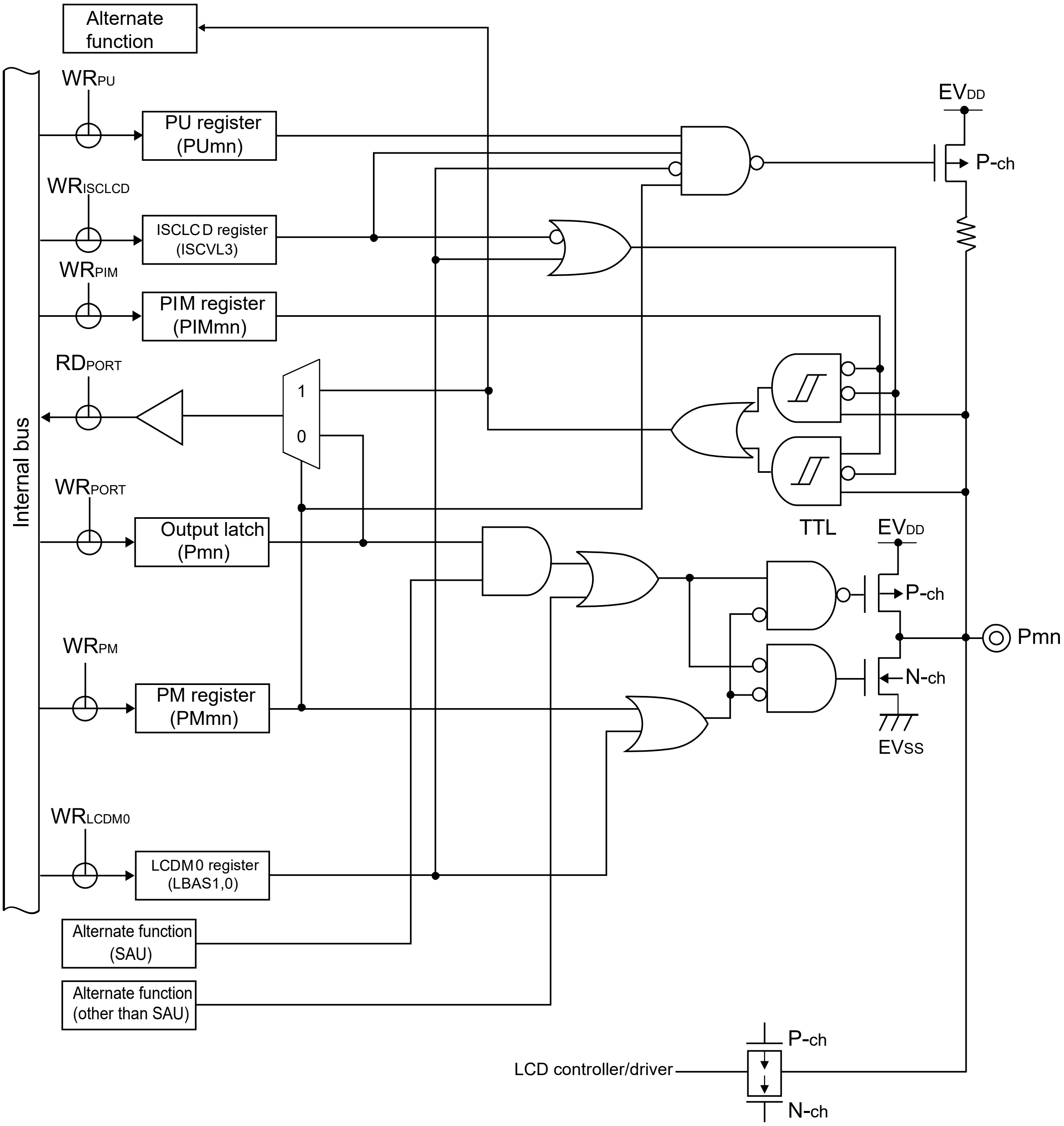


Caution **Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, a through current may flow through in the case of high level input. It is recommended to input a low level to prevent a through current.**

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

2. SAU: Serial array unit

Figure 2-12. Pin Block Diagram for Pin Type 8-5-3

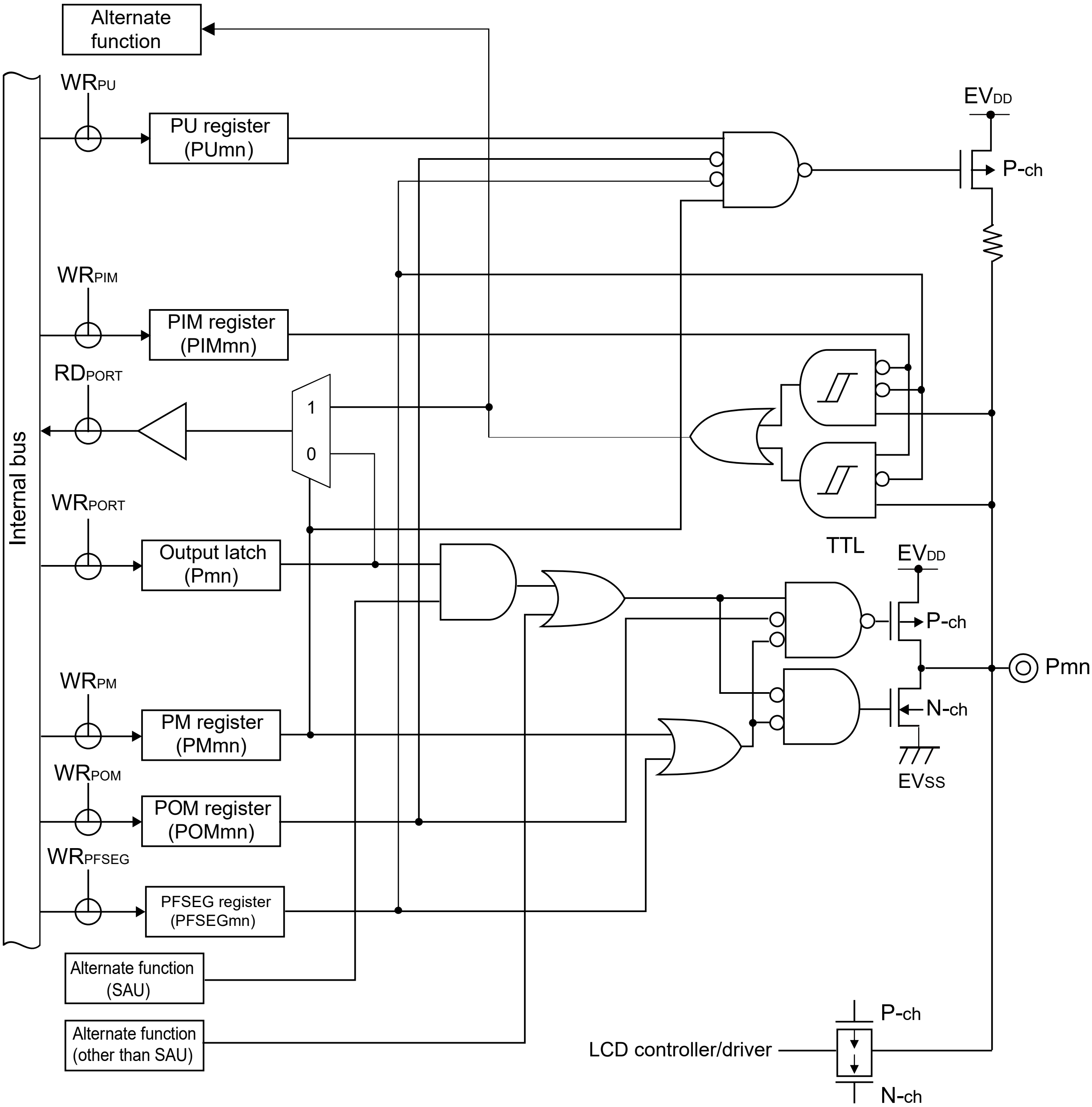


Caution **Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, a through current may flow through in the case of high level input. It is recommended to input a low level to prevent a through current.**

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

2. SAU: Serial array unit

Figure 2-13. Pin Block Diagram for Pin Type 8-5-7



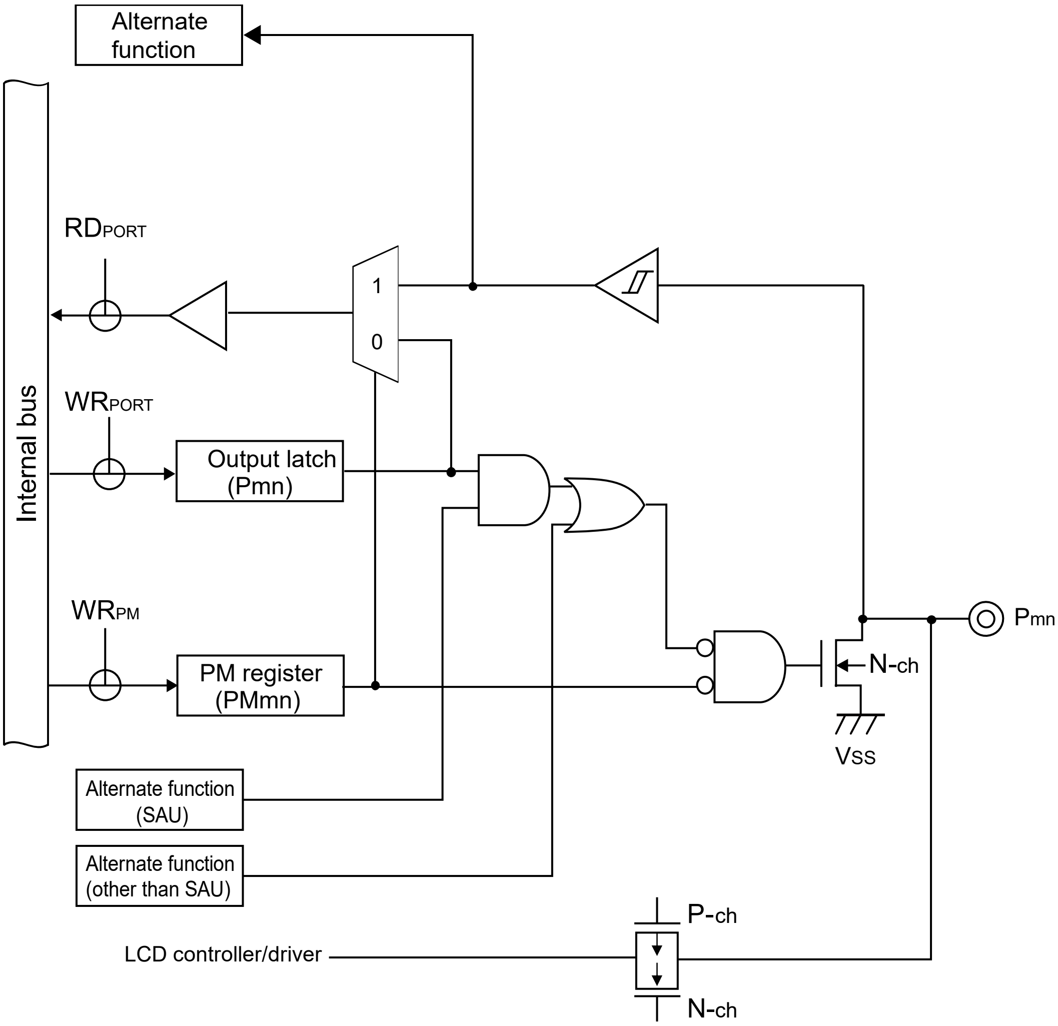
Cautions 1. A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

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.

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

2. SAU: Serial array unit

Figure 2-14. Pin Block Diagram for Pin Type 12-1-4



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 output mode.

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

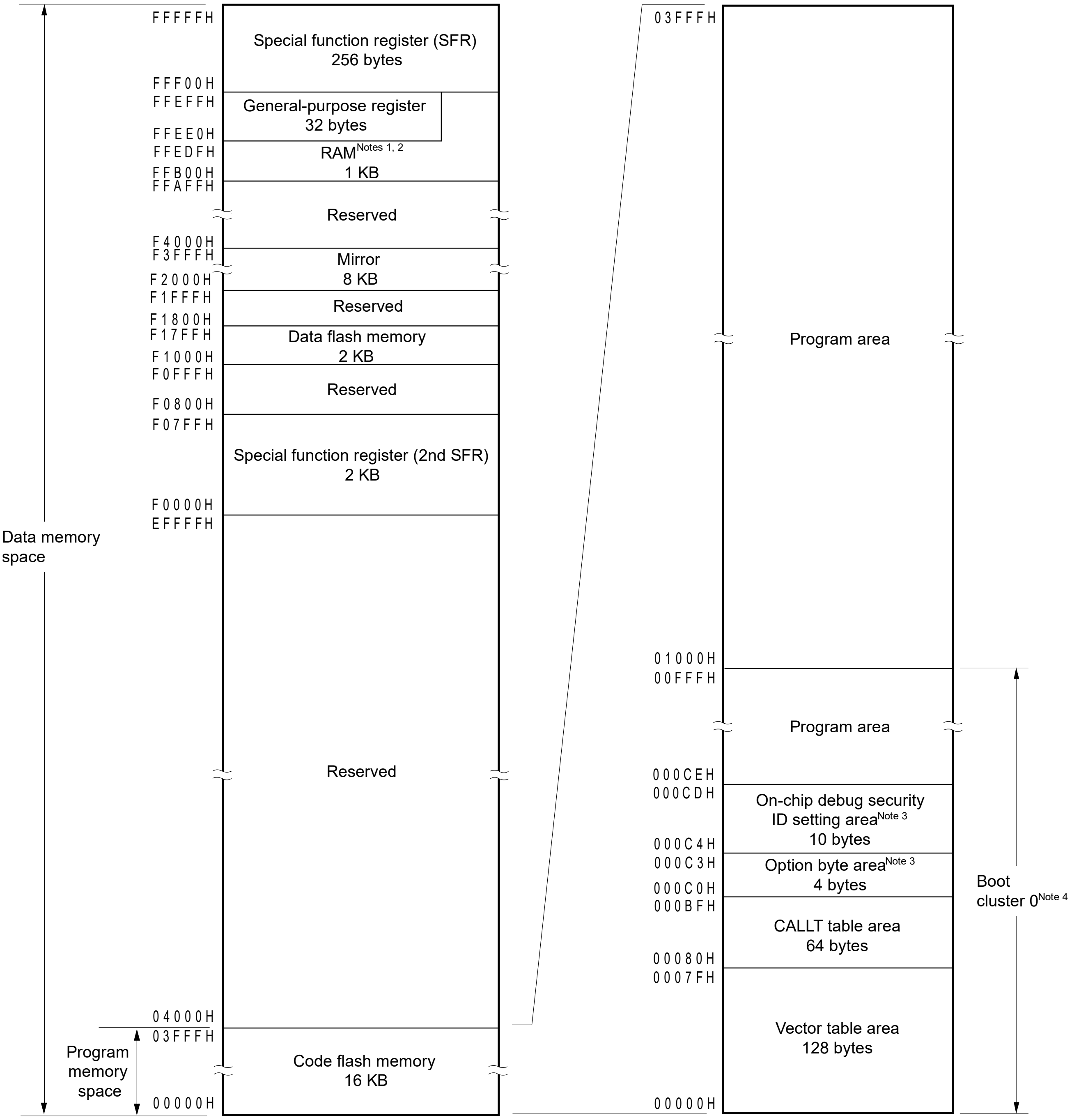
2. SAU: Serial array unit

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the R7F0C001G/L, R7F0C002G/L can access a 1 MB address space. Figures 3-1, 3-2 show the memory maps.

Figure 3-1. Memory Map (R7F0C001G, R7F0C001L)



Notes 1. Use of the area FFE20H to FFEDFH and FFB00H to FFC89H is prohibited when using the self-programming function and data flash function, because this area is used for self-programming library.

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

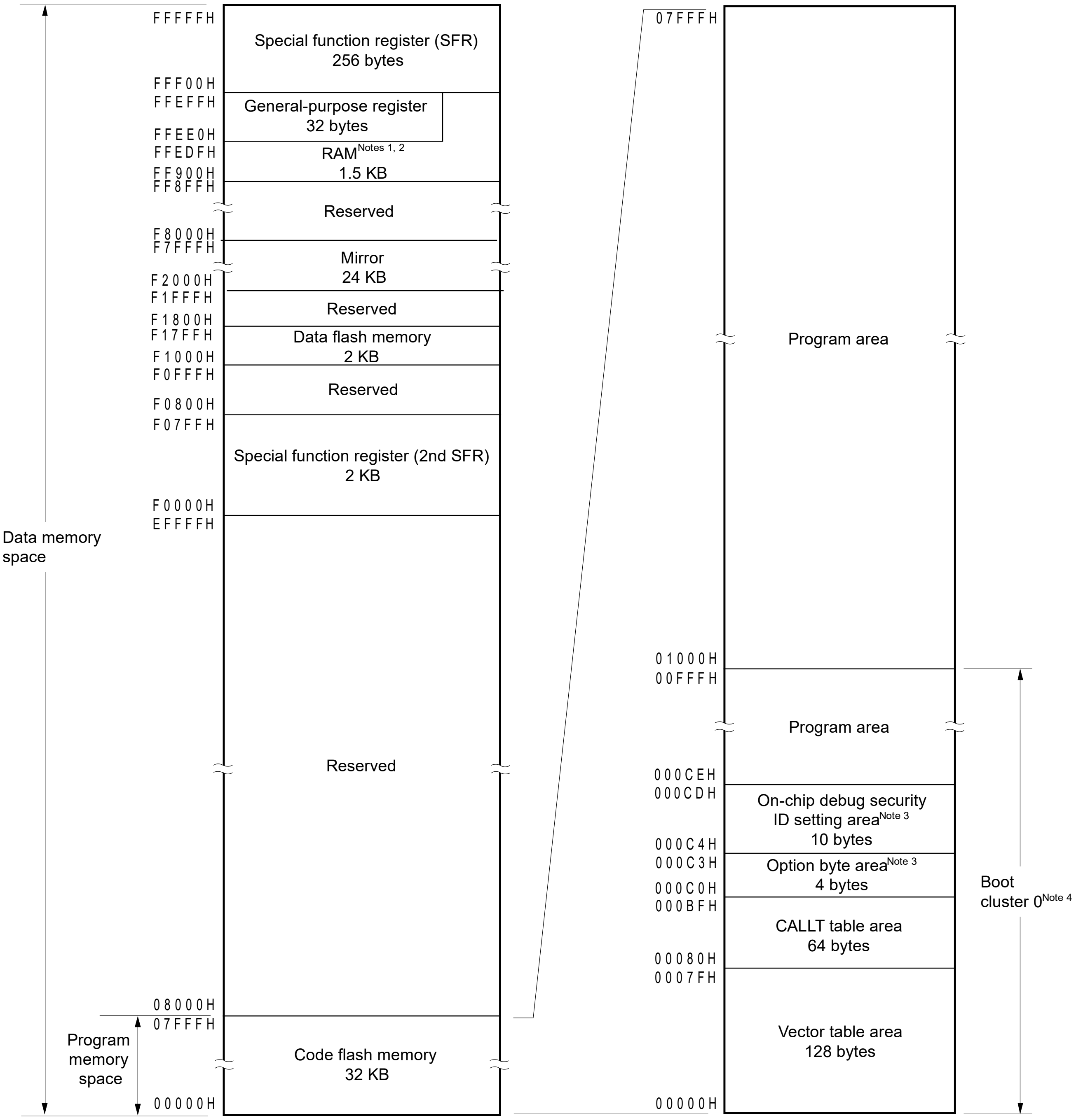
3. Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **23.7 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 20.3.3 RAM parity error detection function.

Figure 3-2. Memory Map (R7F0C002G, R7F0C002L)



Notes 1. Use of the area FFE20H to FFEDFH and FF900H to FFC89H is prohibited when using the self-programming function and data flash function, because this area is used for self-programming library.

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

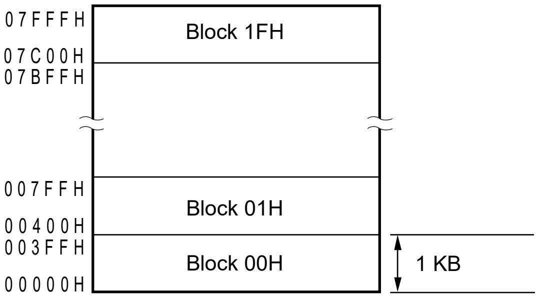
3. Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **23.7 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 20.3.3 RAM parity error detection function.

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



(R7F0C002G, R7F0C002L)

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

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

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
00000H to 003FFH	00H	02000H to 023FFH	08H	04000H to 043FFH	10H	06000H to 063FFH	18H
00400H to 007FFH	01H	02400H to 027FFH	09H	04400H to 047FFH	11H	06400H to 067FFH	19H
00800H to 00BFFH	02H	02800H to 02BFFH	0AH	04800H to 04BFFH	12H	06800H to 06BFFH	1AH
00C00H to 00FFFH	03H	02C00H to 02FFFH	0BH	04C00H to 04FFFH	13H	06C00H to 06FFFH	1BH
01000H to 013FFH	04H	03000H to 033FFH	0CH	05000H to 053FFH	14H	07000H to 073FFH	1CH
01400H to 017FFH	05H	03400H to 037FFH	0DH	05400H to 057FFH	15H	07400H to 077FFH	1DH
01800H to 01BFFH	06H	03800H to 03BFFH	0EH	05800H to 05BFFH	16H	07800H to 07BFFH	1EH
01C00H to 01FFFH	07H	03C00H to 03FFFH	0FH	05C00H to 05FFFH	17H	07C00H to 07FFFH	1FH

Remark R7F0C001G, R7F0C001L : Block numbers 00H to 0FH
R7F0C002G, R7F0C002L: Block numbers 00H to 1FH

3.1.1 Internal program memory space

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

The R7F0C001G/L, R7F0C002G/L products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R7F0C001G, R7F0C001L	Flash memory	16384 × 8 bits (00000H to 03FFFH)
R7F0C002G, R7F0C002L		32768 × 8 bits (00000H to 07FFFH)

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

(1) Vector table area

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

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

Table 3-3. Vector Table

Vector Table Address	Interrupt Source	64-pin	48-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	√	√
00018H	INTST0	√	√
	INTCSI00	√	√
0001AH	INTSR0	√	√
	INTCSI01	√	√
0001CH	INTSRE0	√	√
	INTTM01H	√	√
00020H	INTTM00	√	√
00024H	INTTM03H	√	√
00028H	INTTM01	√	√
0002AH	INTTM02	√	√
0002CH	INTTM03	√	√
0002EH	INTAD	√	√
00030H	INTRTC	√	√
00032H	INTIT	√	√
00034H	INTKR	√	√
00040H	INTTM06	√	√
00042H	INTTM07	√	√
00046H	INTP6	√	–
00048H	INTP7	√	–
0004CH	INTFL	√	√
0007EH	BRK	√	√

Remark √: Mounted
–: Not mounted

(2) CALLT instruction table area

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

(3) Option byte area

A 4-byte area of 000C0H to 000C3H can be used as an option byte area. For details, see **CHAPTER 22 OPTION BYTE**.

(4) On-chip debug security ID setting area

A 10-byte area of 000C4H to 000CDH can be used as an on-chip debug security ID setting area. For details, see **CHAPTER 24 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The R7F0C001G/L, R7F0C002G/L mirrors the code flash area of 02000H to 07FFFH, to F2000H to F7FFFH (the code flash area to be mirrored is set by the processor mode control register (PMC)).

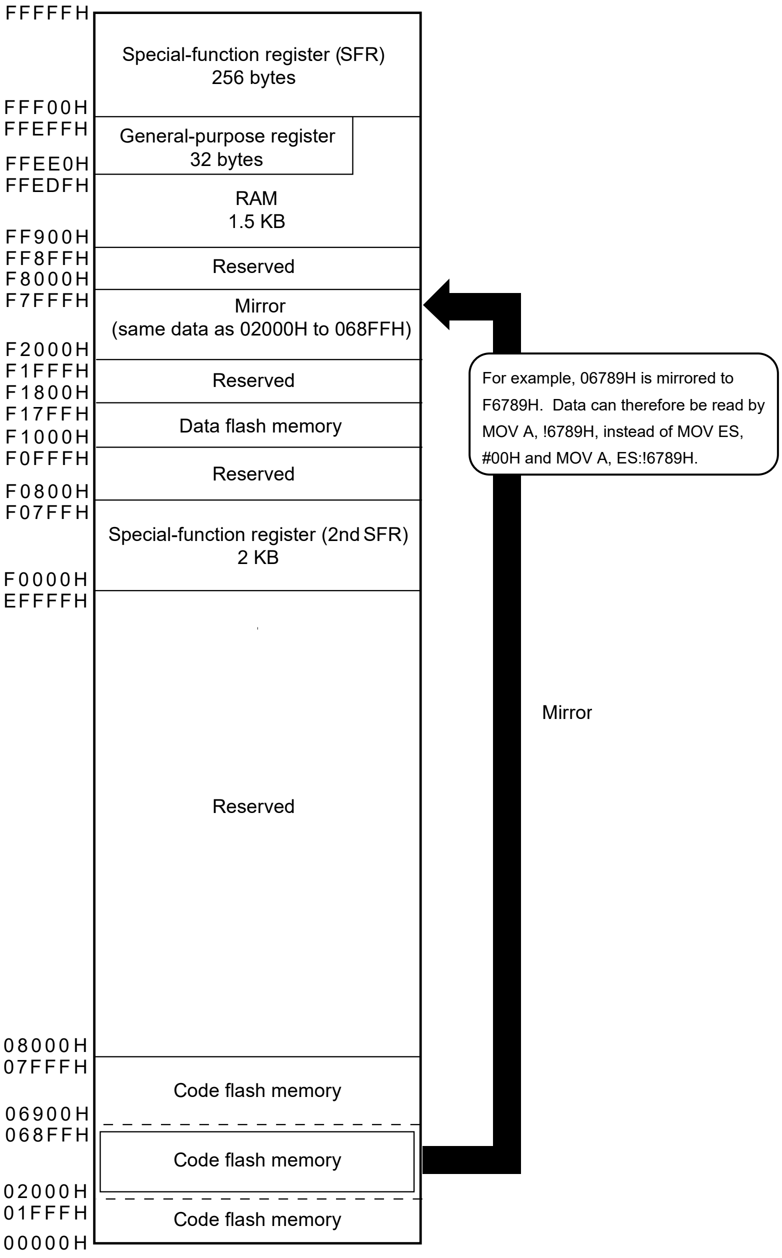
By reading data from F2000H to F7FFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

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 R7F0C002G, R7F0C002L (Flash memory: 32 KB, RAM: 1.5 KB)



For example, 06789H is mirrored to F6789H. Data can therefore be read by MOV A, !6789H, instead of MOV ES, #00H and MOV A, ES:!6789H.

The PMC register is described below.

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

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

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

  Reset signal generation sets this register to 00H.

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

Address: FFFFEH After reset: 00H R/W

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

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

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

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

3.1.3 Internal data memory space

The R7F0C001G/L, R7F0C002G/L products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
R7F0C001G, R7F0C001L	1024 × 8 bits (FFB00H to FFEFFH)
R7F0C002G, R7F0C002L	1536 × 8 bits (FF900H to FFEFFH)

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

The internal RAM is used as stack memory.

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

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

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

R7F0C001G, R7F0C001L : FFB00H to FFC89H
R7F0C002G, R7F0C002L : FF900H to FFC89H

3.1.4 Special function register (SFR) area

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

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

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

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

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

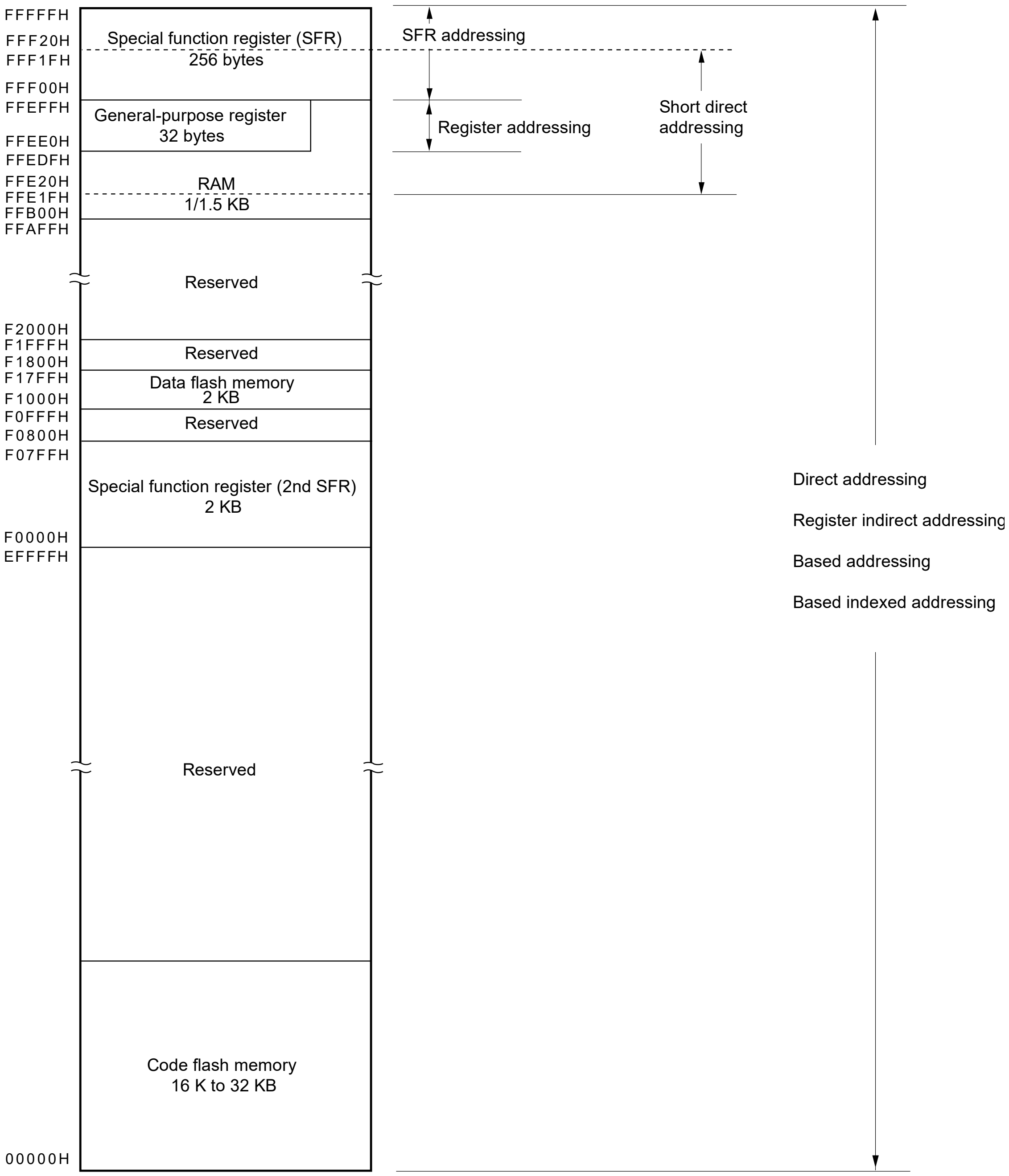
Caution Do not access addresses to which extended SFRs are not assigned.

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 R7F0C001G/L, R7F0C002G/L, based on operability and other considerations. In particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figure 3-4 shows correspondence between data memory and addressing. For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3-4 Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The R7F0C001G/L, R7F0C002G/L 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)

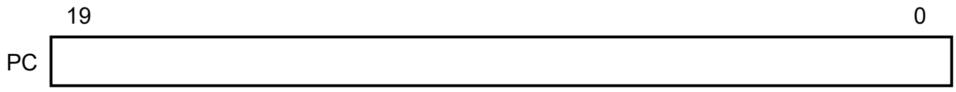
The program counter is a 20-bit register that holds the address information of the next program to be executed.

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 3-5. Format of Program Counter

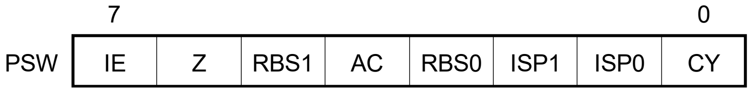


(2) Program status word (PSW)

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 3-6. Format of Program Status Word



(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

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

This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L) (see **14.3.3**) can not be acknowledged. Actual vectored interrupt 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 3-7. Format of Stack Pointer

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

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.

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

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

3. The internal RAM in the following products cannot be used as stack area when using the self-programming function and data flash function.

R7F0C001G, R7F0C001L :	**FFB00H to FFC89H**
R7F0C002G, R7F0C002L:	**FF900H to FFC89H**

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).

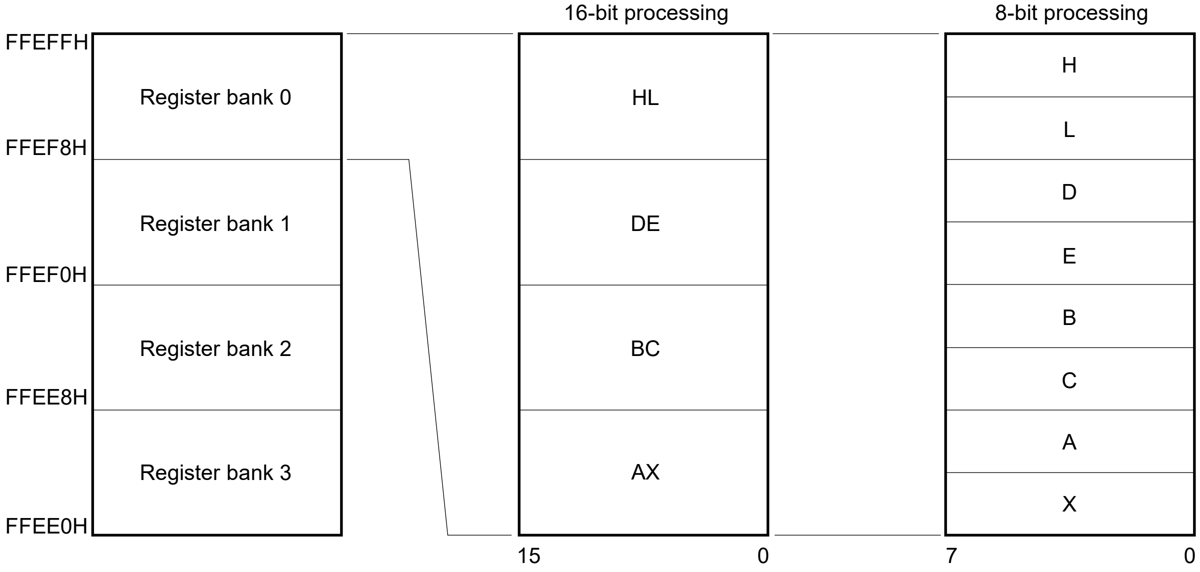
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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 interrupt processing 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 3-8. 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.

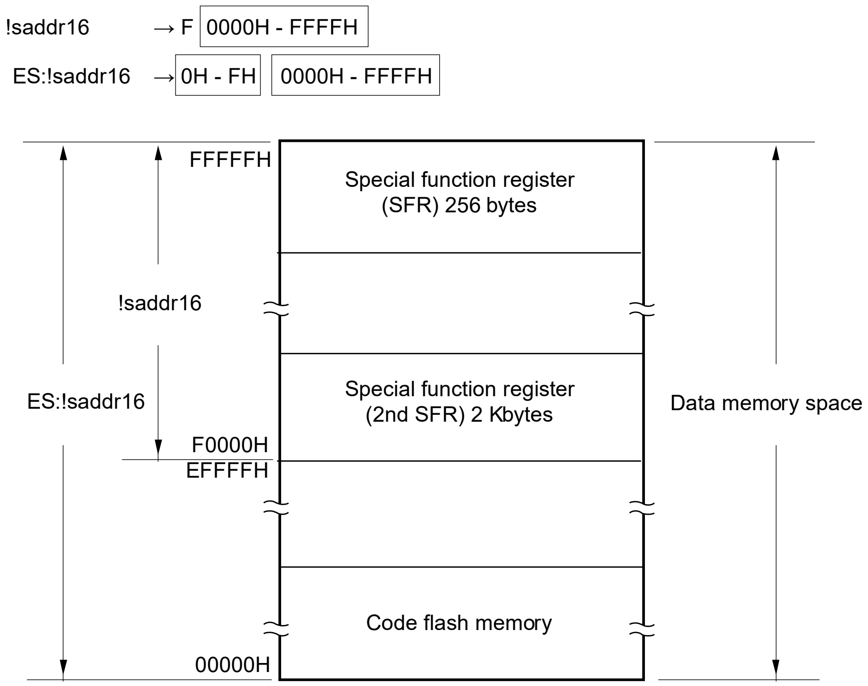
Figure 3-9. Configuration of ES and CS Registers

	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 3-10. Extension of Data Area Which Can Be Accessed



3.2.4 Special function registers (SFRs)

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

SFRs are allocated to the FFF00H to FFFFFH area.

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

Each manipulation bit unit can be specified as follows.

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

Table 3-5 gives a list of the SFRs. The meanings of items in the table are as follows.

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

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

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF01H	Port register 1	P1		R/W	√	√	–	00H
FFF02H	Port register 2	P2		R/W	√	√	–	00H
FFF03H	Port register 3	P3		R/W	√	√	–	00H
FFF04H	Port register 4	P4		R/W	√	√	–	00H
FFF05H	Port register 5	P5		R/W	√	√	–	00H
FFF06H	Port register 6	P6		R/W	√	√	–	00H
FFF07H	Port register 7	P7		R/W	√	√	–	00H
FFF0CH	Port register 12	P12		R/W	√	√	–	Undefined
FFF0DH	Port register 13	P13		R/W	√	√	–	Undefined
FFF0EH	Port register 14	P14		R/W	√	√	–	00H
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	–	√	√	0000H
FFF11H		–			–	–		
FFF12H	Serial data register 01	RXD0/ SIO01	SDR01	R/W	–	√	√	0000H
FFF13H		–			–	–		
FFF18H	Timer data register 00	TDR00		R/W	–	–	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	–	√	√	00H
FFF1BH		TDR01H			–	√		00H
FFF1EH	10-bit A/D conversion result register	ADCR		R	–	–	√	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	–	√	–	00H
FFF21H	Port mode register 1	PM1		R/W	√	√	–	FFH
FFF22H	Port mode register 2	PM2		R/W	√	√	–	FFH
FFF23H	Port mode register 3	PM3		R/W	√	√	–	FFH
FFF24H	Port mode register 4	PM4		R/W	√	√	–	FFH
FFF25H	Port mode register 5	PM5		R/W	√	√	–	FFH
FFF26H	Port mode register 6	PM6		R/W	√	√	–	FFH
FFF27H	Port mode register 7	PM7		R/W	√	√	–	FFH
FFF2CH	Port mode register 12	PM12		R/W	√	√	–	FFH
FFF2EH	Port mode register 14	PM14		R/W	√	√	–	FFH
FFF30H	A/D converter mode register 0	ADM0		R/W	√	√	–	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	–	00H
FFF32H	A/D converter mode register 1	ADM1		R/W	√	√	–	00H
FFF34H	Key return control register	KRCTL		R/W	√	√	–	00H
FFF35H	Key return flag register	KRF		R/W	–	√	–	00H
FFF37H	Key return mode register 0	KRM0		R/W	√	√	–	00H

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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	LCDC0		R/W	–	√	–	00H
FFF43H	LCD boost level control register	VLCD		R/W	–	√	–	04H
FFF64H	Timer data register 02	TDR02		R/W	–	–	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	–	√	√	00H
FFF67H		TDR03H			–	√		00H
FFF6CH	Timer data register 06	TDR06		R/W	–	–	√	0000H
FFF6DH								
FFF6EH	Timer data register 07	TDR07		R/W	–	–	√	0000H
FFF6FH								
FFF90H	Interval timer control register	ITMC		R/W	–	–	√	0FFFH
FFF91H								
FFF92H	Second count register	SEC		R/W	–	√	–	00H
FFF93H	Minute count register	MIN		R/W	–	√	–	00H
FFF94H	Hour count register	HOUR		R/W	–	√	–	12H[Note]
FFF95H	Week count register	WEEK		R/W	–	√	–	00H
FFF96H	Day count register	DAY		R/W	–	√	–	01H
FFF97H	Month count register	MONTH		R/W	–	√	–	01H
FFF98H	Year count register	YEAR		R/W	–	√	–	00H
FFF99H	Watch error correction register	SUBCUD		R/W	–	√	–	00H
FFF9AH	Alarm minute register	ALARMWM		R/W	–	√	–	00H
FFF9BH	Alarm hour register	ALARMWH		R/W	–	√	–	12H
FFF9CH	Alarm week register	ALARMWW		R/W	–	√	–	00H
FFF9DH	Real-time clock control register 0	RTCC0		R/W	√	√	–	00H
FFF9EH	Real-time clock control register 1	RTCC1		R/W	√	√	–	00H
FFFA0H	Clock operation mode control register	CMC		R/W	–	√	–	00H
FFFA1H	Clock operation status control register	CSC		R/W	√	√	–	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	√	√	–	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	–	√	–	07H
FFFA4H	System clock control register	CKC		R/W	√	√	–	00H

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

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
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 1
FFFA9H	Voltage detection register	LVIM	R/W	√	√	–	00H Note 1
FFFAAH	Voltage detection level register	LVIS	R/W	√	√	–	00H/01H/81H Note 1
FFFABH	Watchdog timer enable register	WDTE	R/W	–	√	–	1AH/9AH Note 2
FFFACH	CRC input register	CRCIN	R/W	–	√	–	00H

Notes 1. The reset values of the registers vary depending on the reset source as shown below.

<table>
<tr><th colspan="2">Reset Source
Register</th><th>$\overline{\text{RESET}}$ Input</th><th>Reset by POR</th><th>Reset by Execution of Illegal Instruction</th><th>Reset by WDT</th><th>Reset by RAM parity error</th><th>Reset by illegal-memory access</th><th>Reset by LVD</th></tr>
<tr><td rowspan="5">RESF</td><td>TRAP bit</td><td colspan="2" rowspan="5">Cleared (0)</td><td>Set (1)</td><td colspan="3">Held</td><td rowspan="4">Held</td></tr>
<tr><td>WDTRF bit</td><td>Held</td><td>Set (1)</td><td colspan="2">Held</td></tr>
<tr><td>RPERF bit</td><td colspan="2">Held</td><td>Set (1)</td><td>Held</td></tr>
<tr><td>IAWRF bit</td><td colspan="3">Held</td><td>Set (1)</td></tr>
<tr><td>LVIRF bit</td><td colspan="4">Held</td><td>Set (1)</td></tr>
<tr><td rowspan="3">LVIM</td><td>LVISEN bit</td><td colspan="6">Cleared (0)</td><td rowspan="4">Held</td></tr>
<tr><td>LVIOMSK bit</td><td colspan="6" rowspan="2">Held</td></tr>
<tr><td>LVIF bit</td></tr>
<tr><td colspan="2">LVIS</td><td colspan="6">Cleared (00H/01H/81H)</td></tr>
</table>

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
FFFE0H	Interrupt request flag register 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	Interrupt request flag register 0H	IF0H		R/W	√	√		00H
FFFE2H	Interrupt request flag register 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	Interrupt request flag register 1H	IF1H		R/W	√	√		00H
FFFE4H	Interrupt mask flag register 0L	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H	Interrupt mask flag register 0H	MK0H		R/W	√	√		FFH
FFFE6H	Interrupt mask flag register 1L	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H	Interrupt mask flag register 1H	MK1H		R/W	√	√		FFH
FFFE8H	Priority specification flag register 00L	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H	Priority specification flag register 00H	PR00H		R/W	√	√		FFH
FFFEAH	Priority specification flag register 01L	PR01L	PR01	R/W	√	√	√	FFH
FFFEBH	Priority specification flag register 01H	PR01H		R/W	√	√		FFH
FFFECH	Priority specification flag register 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFEDH	Priority specification flag register 10H	PR10H		R/W	√	√		FFH
FFFEEH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	Priority specification flag register 11H	PR11H		R/W	√	√		FFH
FFFFEH	Processor mode control register	PMC		R/W	√	√	–	00H

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

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

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

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

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

Each manipulation bit unit can be specified as follows.

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

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

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

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

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

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

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
F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
F0033H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	–	01H
F0035H	Pull-up resistor option register 5	PU5	R/W	√	√	–	00H
F0037H	Pull-up resistor option register 7	PU7	R/W	√	√	–	00H
F003CH	Pull-up resistor option register 12	PU12	R/W	√	√	–	00H
F003EH	Pull-up resistor option register 14	PU14	R/W	√	√	–	00H
F0041H	Port input mode register 1	PIM1	R/W	√	√	–	00H
F0051H	Port output mode register 1	POM1	R/W	√	√	–	00H
F0061H	Port mode control register 1	PMC1	R/W	√	√	–	FFH
F0064H	Port mode control register 4	PMC4	R/W	√	√	–	FFH
F006CH	Port mode control register 12	PMC12	R/W	√	√	–	FFH
F006EH	Port mode control register 14	PMC14	R/W	√	√	–	FFH
F0070H	Noise filter enable register 0	NFEN0	R/W	√	√	–	00H
F0071H	Noise filter enable register 1	NFEN1	R/W	√	√	–	00H
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
F0090H	Data flash control register	DFLCTL	R/W	√	√	–	00H
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV	R/W	–	√	–	Undefined[Note]

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F00F0H	Peripheral enable register 0	PER0		R/W	√	√	–	00H
F00F3H	Subsystem clock supply mode control register	OSMC		R/W	–	√	–	00H
F00F5H	RAM parity error control register	RPECTL		R/W	√	√	–	00H
F00FEH	BCD adjust result register	BCDADJ		R	–	√	–	Undefined
F0100H	Serial status register 00	SSR00L	SSR00	R	–	√	√	0000H
F0101H		–			–	–		
F0102H	Serial status register 01	SSR01L	SSR01	R	–	√	√	0000H
F0103H		–			–	–		
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		–			–	–		
F0110H	Serial mode register 00	SMR00		R/W	–	–	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	–	–	√	0020H
F0113H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F011BH								
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	–	–	√	0000H
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		–			–	–		

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
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							
F018CH	Timer counter register 06	TCR06	R	–	–	√	FFFFH
F018DH							
F018EH	Timer counter register 07	TCR07	R	–	–	√	FFFFH
F018FH							
F0190H	Timer mode register 00	TMR00	R/W	–	–	√	0000H
F0191H							
F0192H	Timer mode register 01	TMR01	R/W	–	–	√	0000H
F0193H							
F0194H	Timer mode register 02	TMR02	R/W	–	–	√	0000H
F0195H							
F0196H	Timer mode register 03	TMR03	R/W	–	–	√	0000H
F0197H							
F019CH	Timer mode register 06	TMR06	R/W	–	–	√	0000H
F019DH							
F019EH	Timer mode register 07	TMR07	R/W	–	–	√	0000H
F019FH							

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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		–			–	–		
F01ACH	Timer status register 06	TSR06L	TSR06	R	–	√	√	0000H
F01ADH		–			–	–		
F01AEH	Timer status register 07	TSR07L	TSR07	R	–	√	√	0000H
F01AFH		–			–	–		
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H
F01B1H		–			–	–		
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	√	√	√	0000H
F01B3H		–			–	–		
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		–			–	–		
F01B6H	Timer clock select register 0	TPS0		R/W	–	–	√	0000H
F01B7H								
F01B8H	Timer output register 0	TO0L	TO0	R/W	–	√	√	0000H
F01B9H		–			–	–		
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BBH		–			–	–		
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	–	√	√	0000H
F01BDH		–			–	–		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	–	√	√	0000H
F01BFH		–			–	–		
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

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
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	√	√	–	7FH
F0308H	LCD Input switch control register	ISCLCD	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

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

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

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

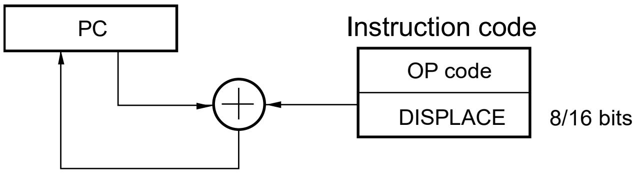
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

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

Figure 3-11. Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

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

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

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

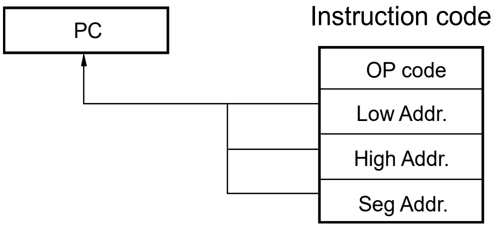
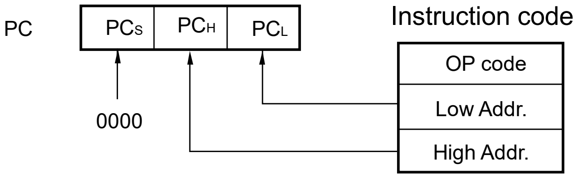


Figure 3-13. Example of CALL !addr16/BR !addr16



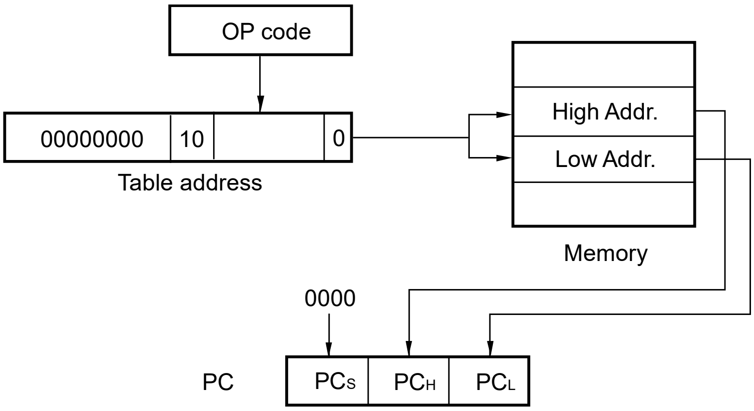
3.3.3 Table indirect addressing

[Function]

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

In the R7F0C001G/L, R7F0C002G/L, branching is enabled only to the 64 KB space from 00000H to 0FFFFH.

Figure 3-14. Outline of Table Indirect Addressing

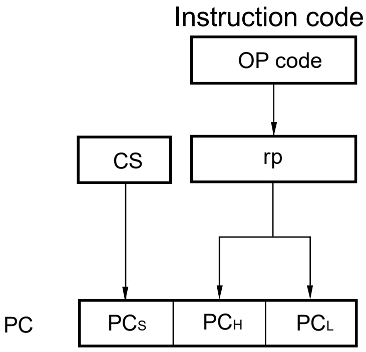


3.3.4 Register direct addressing

[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 3-15. Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

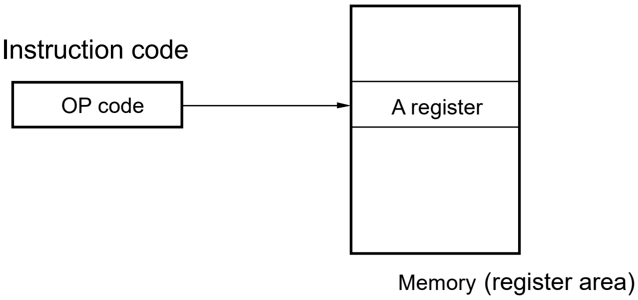
[Function]

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]

Implied addressing can be applied only to MULU X.

Figure 3-16. Outline of Implied Addressing



3.4.2 Register addressing

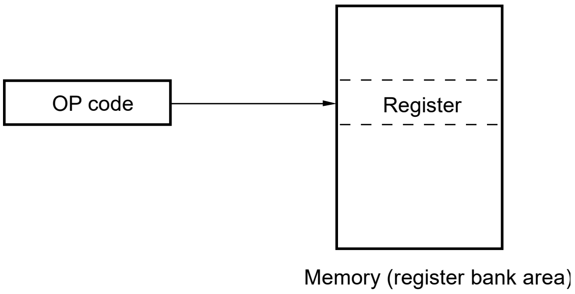
[Function]

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

[Operand format]

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

Figure 3-17. Outline of Register Addressing



3.4.3 Direct addressing

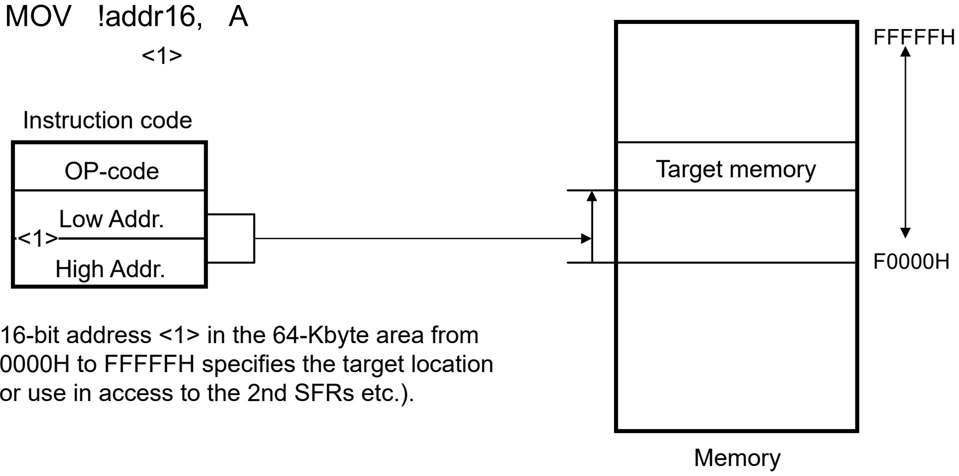
[Function]

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

[Operand format]

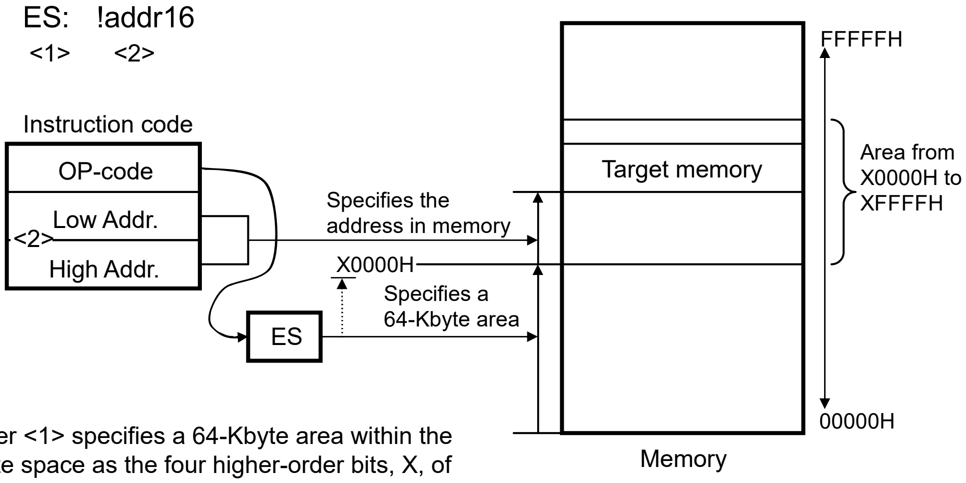
Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3-18. Example of ADDR16



- 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 3-19. 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

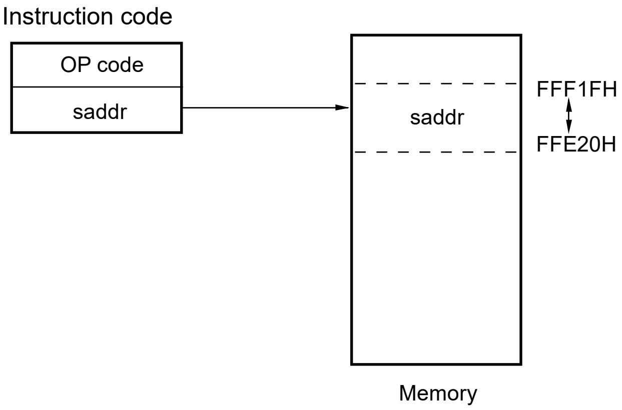
[Function]

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 3-20. Outline of Short Direct Addressing



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

3.4.5 SFR addressing

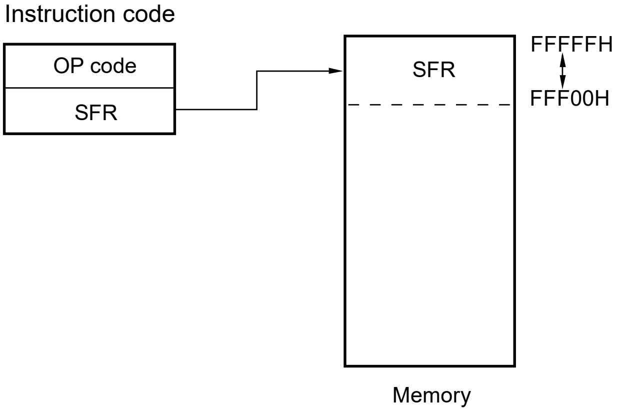
[Function]

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

[Operand format]

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

Figure 3-21. Outline of SFR Addressing



3.4.6 Register indirect addressing

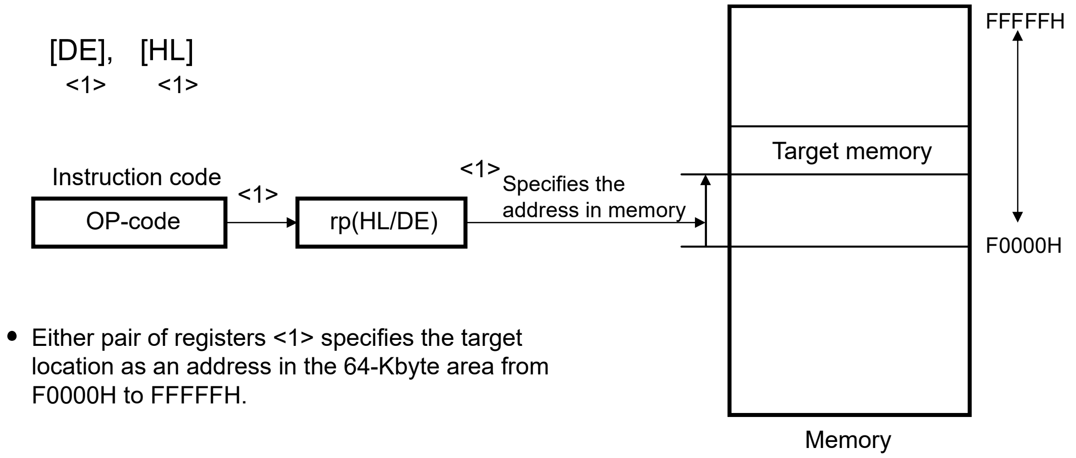
[Function]

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

[Operand format]

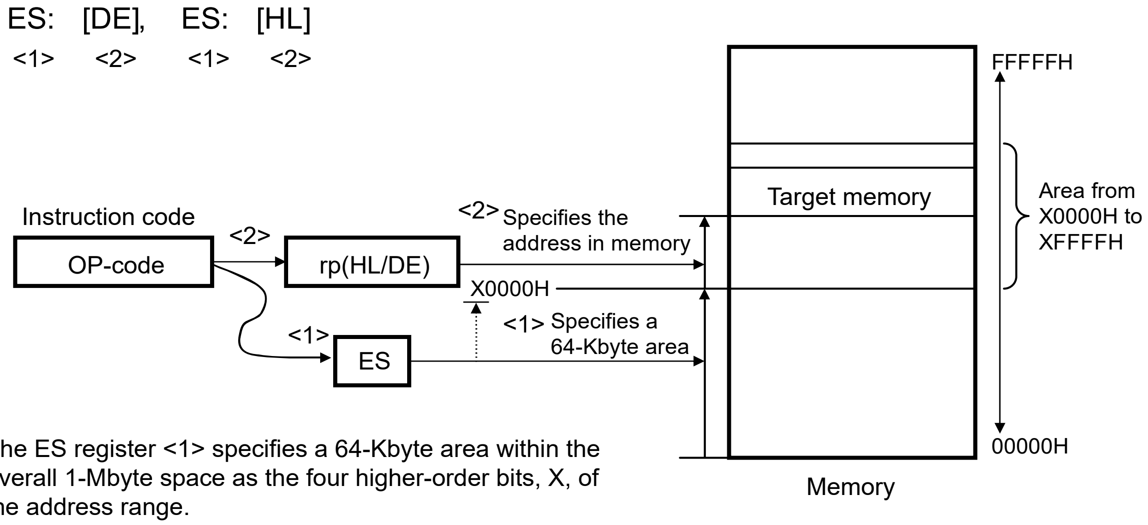
Identifier	Description
–	[DE], [HL] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

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



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

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



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

3.4.7 Based addressing

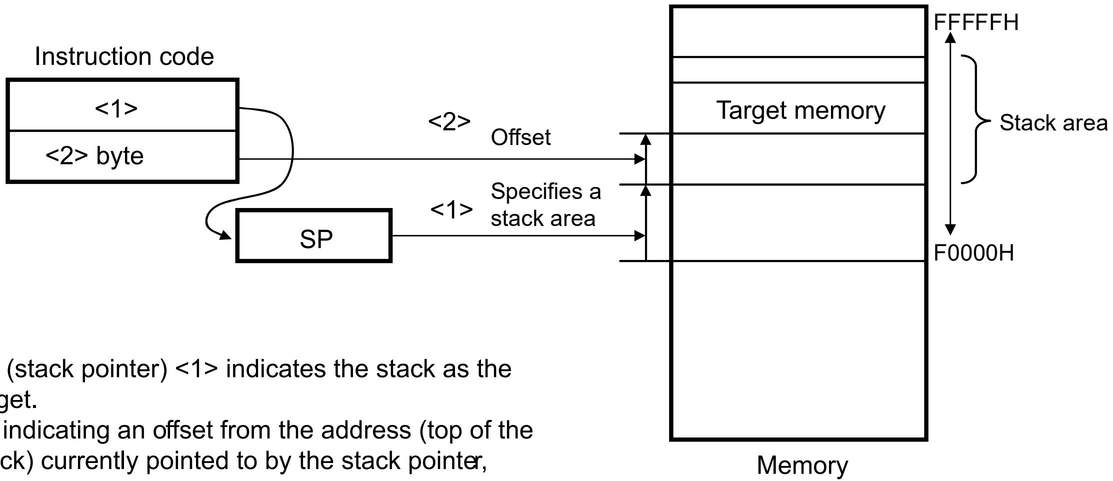
[Function]

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

[Operand format]

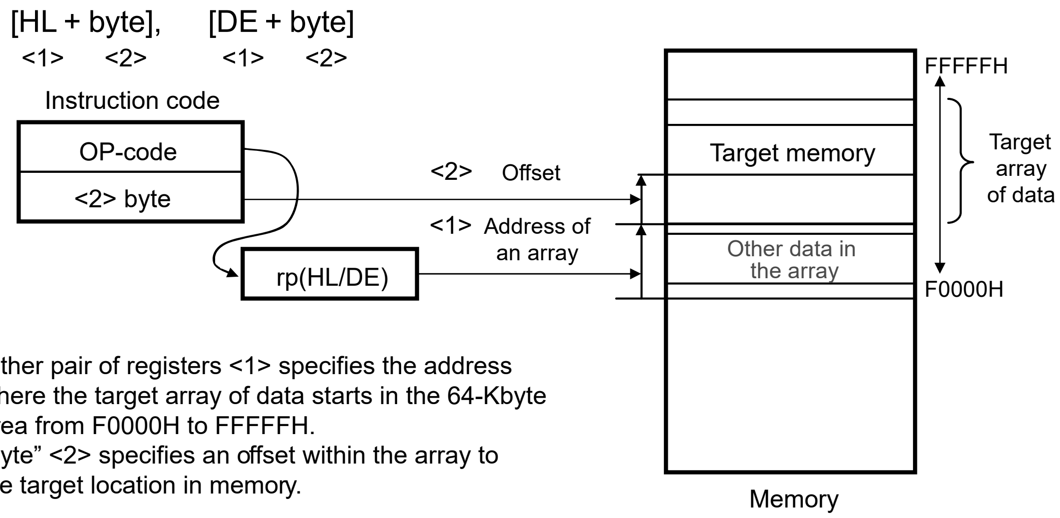
Identifier	Description
–	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
–	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
–	word[BC] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL + byte], ES:[DE + byte] (higher 4-bit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

Figure 3-24. Example of [SP+byte]



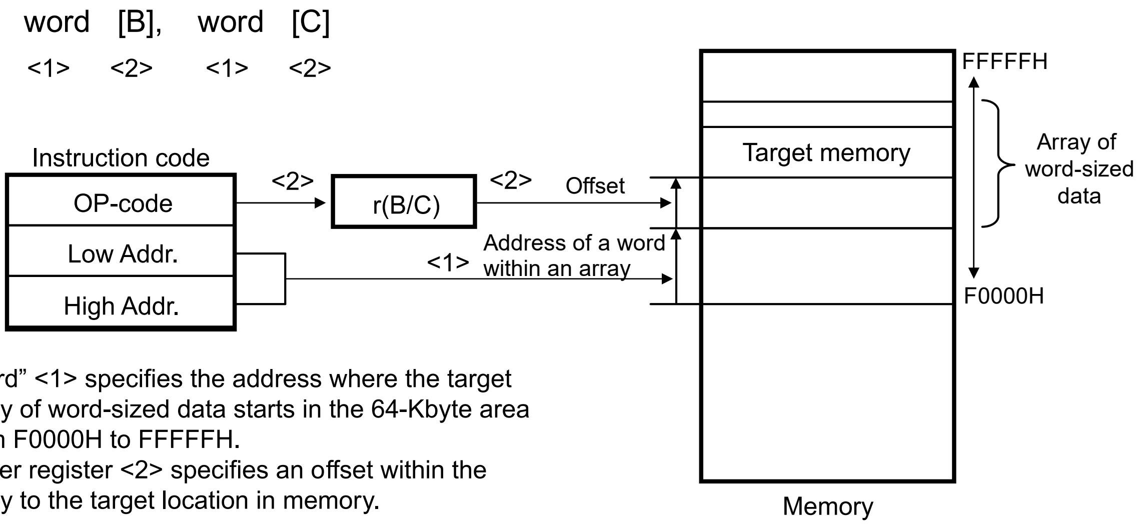
- SP (stack pointer) <1> indicates the stack as the target.
- By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, “byte” <2> indicates the target memory (SP + byte).

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



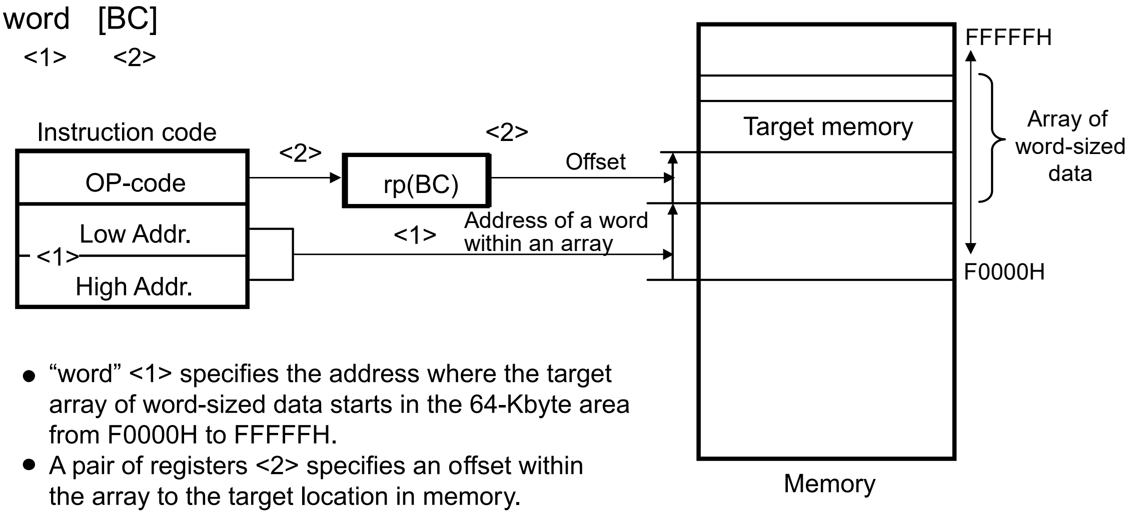
- Either pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- “byte” <2> specifies an offset within the array to the target location in memory.

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



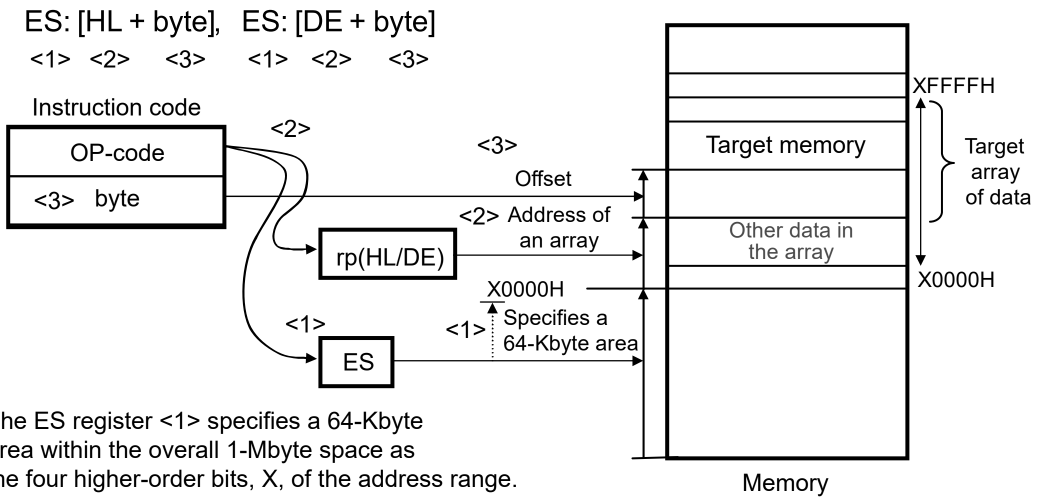
- “word” <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3-27. Example of word[BC]



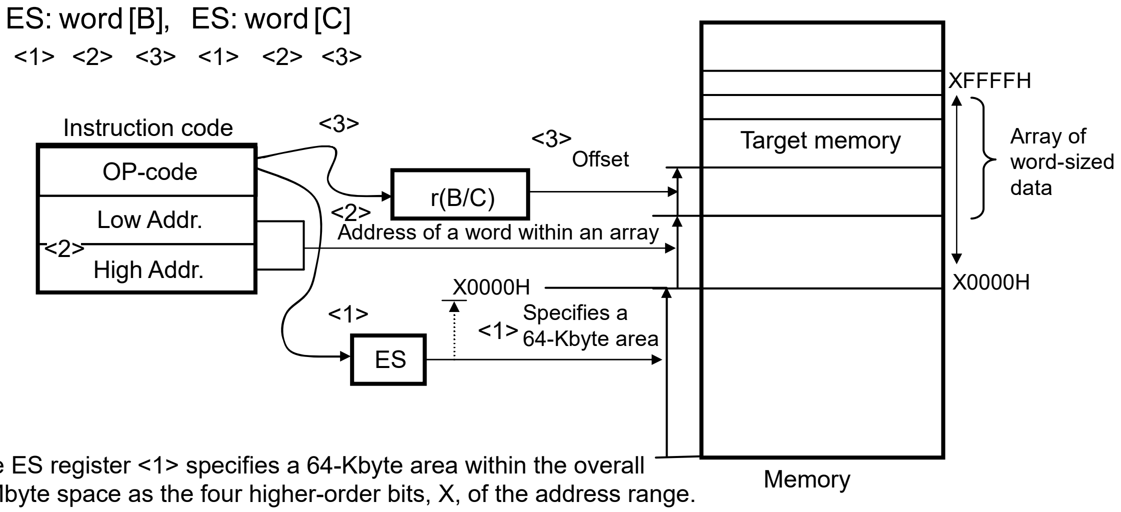
- “word” <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

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



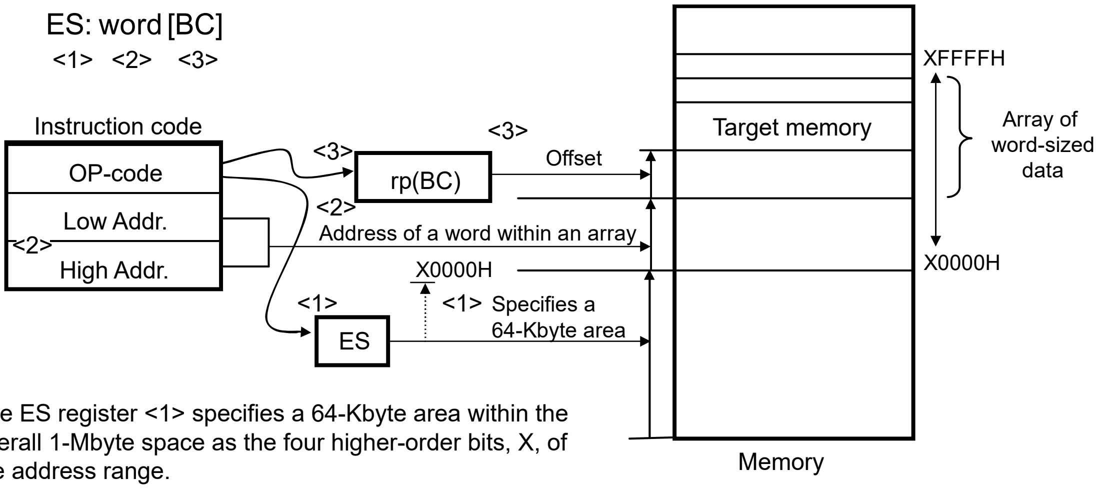
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- “byte” <3> specifies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

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



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

Figure 3-30. Example of ES:word[BC]



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

3.4.8 Based indexed addressing

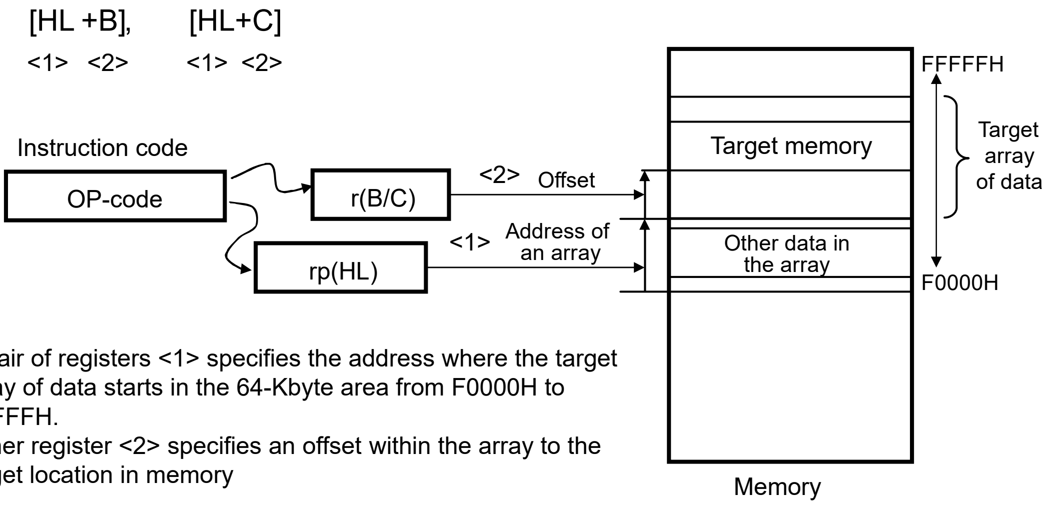
[Function]

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

[Operand format]

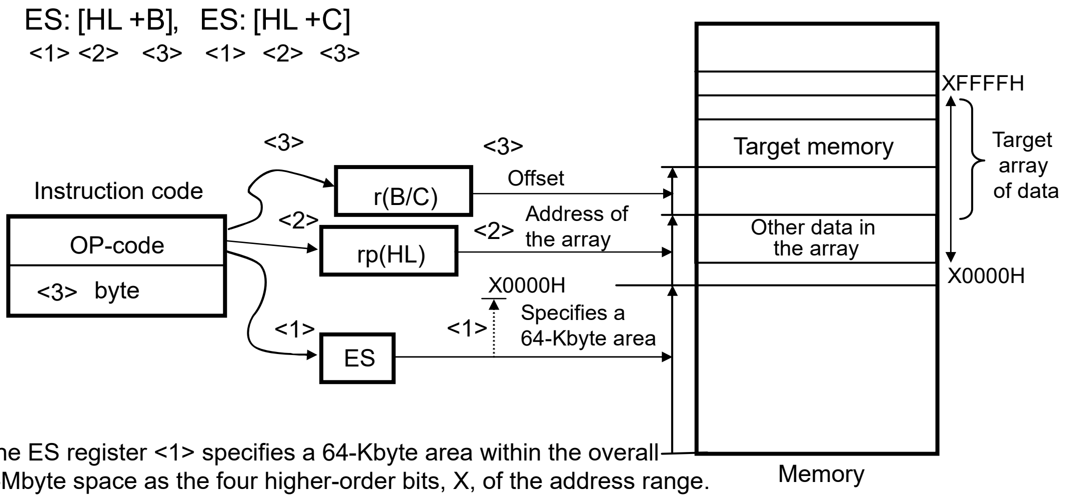
Identifier	Description
–	[HL+B], [HL+C] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL+B], ES:[HL+C] (higher 4-bit addresses are specified by the ES register)

Figure 3-31. Example of [HL+B], [HL+C]



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

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



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

3.4.9 Stack addressing

[Function]

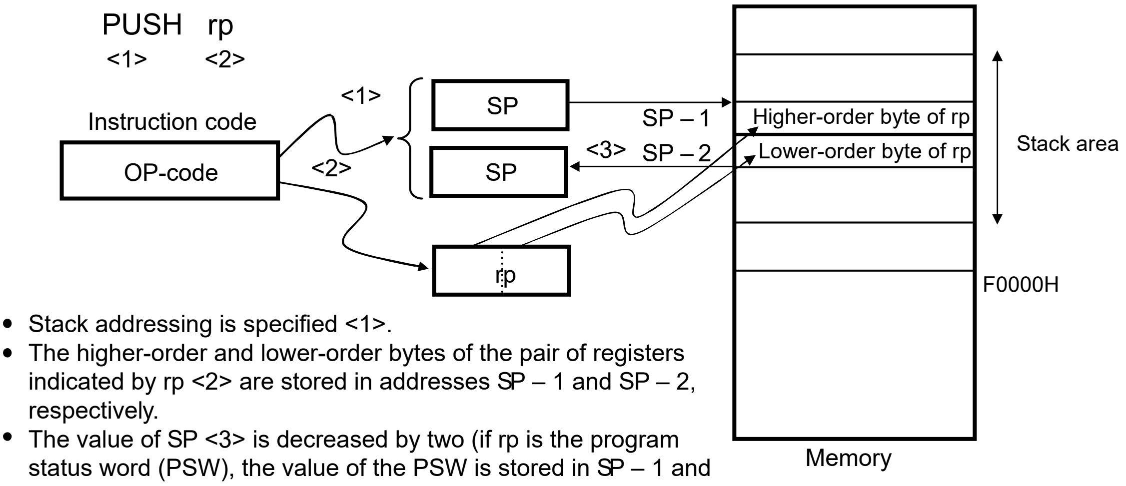
The stack area is indirectly addressed with the stack pointer (SP) contents. This addressing is automatically employed when the PUSH, POP, subroutine call, and return instructions are executed or the register is saved/restored upon generation of an interrupt request.

Stack addressing is applied only to the internal RAM area.

[Description format]

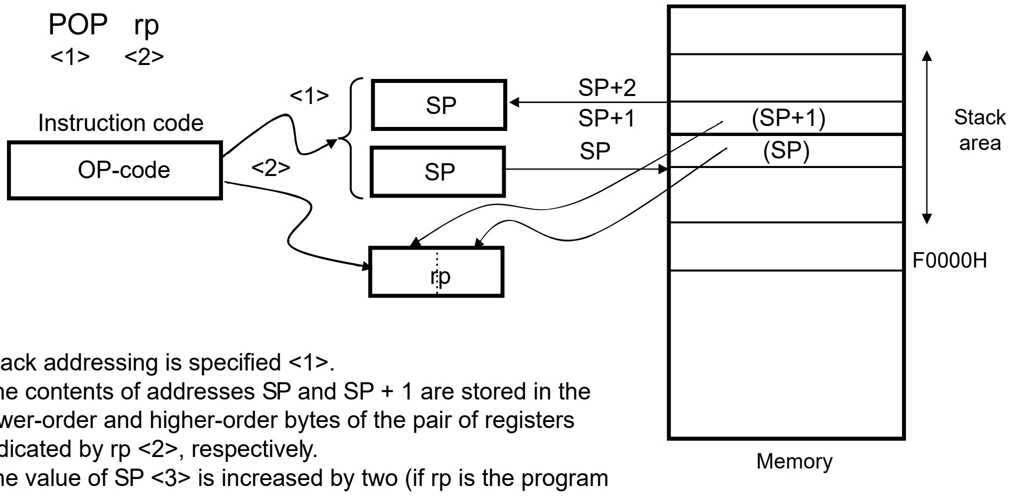
Identifier	Description
–	PUSH PSW AX/BC/DE/HL POP PSW AX/BC/DE/HL CALL/CALLT RET BRK RETB (Interrupt request generated) RETI

Figure 3-33. Example of PUSH rp



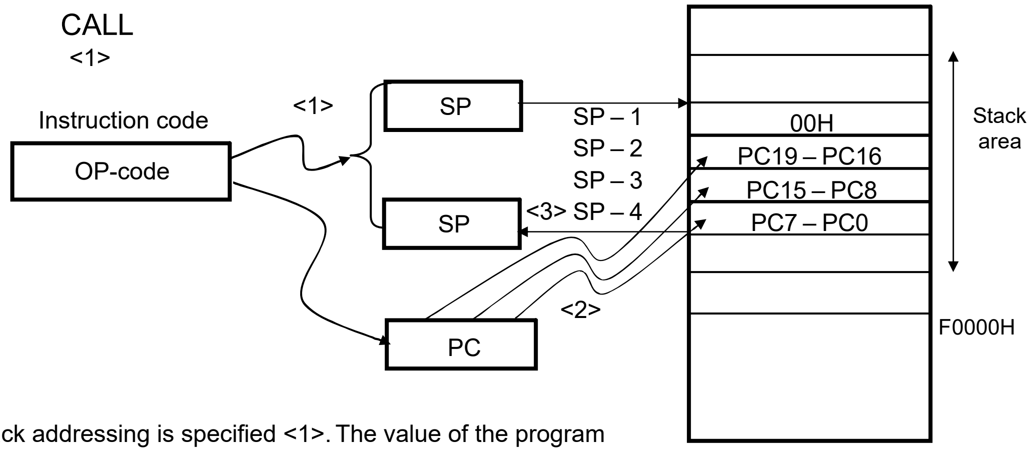
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP – 1 and SP – 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP –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

Figure 3-34. Example of POP



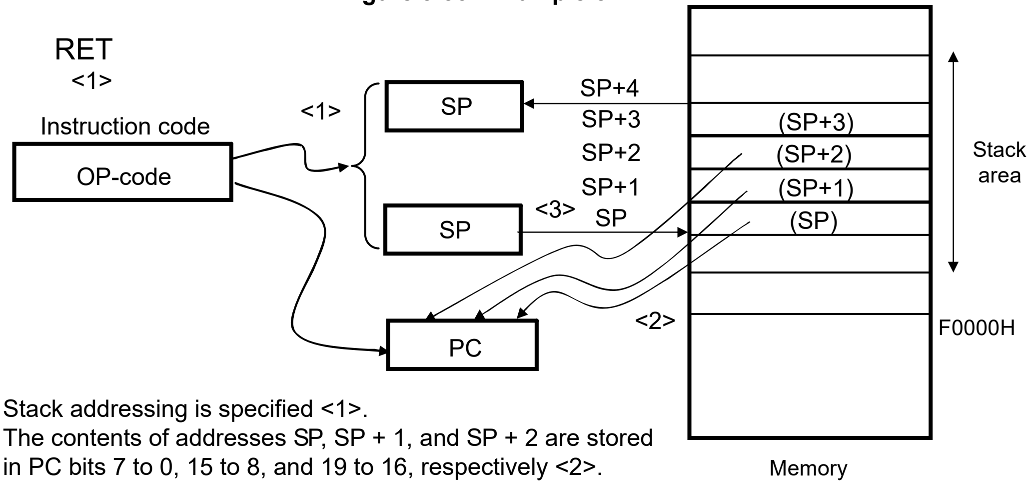
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP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

Figure 3-35. Example of CALL, CALLT



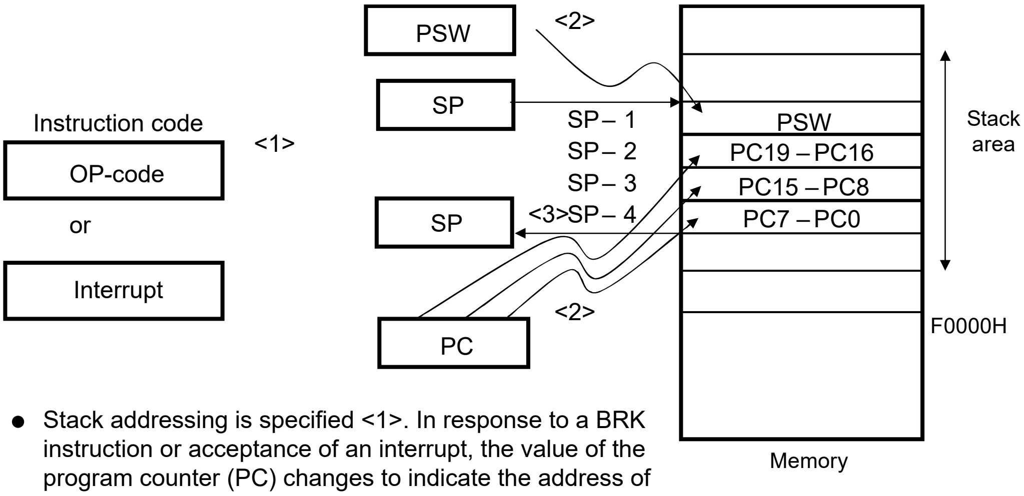
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP – 1, SP – 2, SP – 3, and SP – 4, respectively <2>.
- The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

Figure 3-36. Example of RET



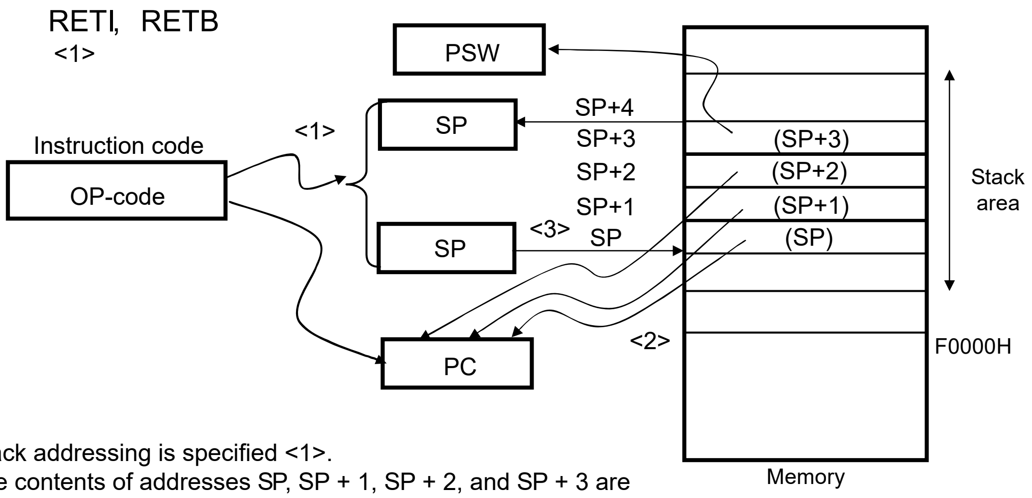
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3-37. Example of Interrupt, BRK



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

Figure 3-38. Example of RETI, RETB



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

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The R7F0C001G/L, R7F0C002G/L 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 (PM1 to PM7, PM12, PM14) Port registers (P1 to P7, P12-P14) Pull-up resistor option registers (PU1, PU3 to PU5, PU7, PU12, PU14) Port input mode register (PIM1) Port output mode register (POM1) Port mode control registers (PMC1, PMC4, PMC12, PMC14) A/D port configuration register (ADPC) Peripheral I/O redirection register (PIOR) LCD port function registers (PFSEG0 PFSEG4) LCD input switch control register (ISCLCD)
Port	● 48-pin products Total: 33 (CMOS I/O: 26, CMOS input: 5, N-ch open drain I/O: 2) ● 64-pin products Total: 47 (CMOS I/O: 39, CMOS input: 5, CMOS output: 1, N-ch open drain I/O: 2)
Pull-up resistor	● 48-pin products Total: 24 ● 64-pin products Total: 37

4.2.1 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, P11, P15, and P16 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, P12, P15, and P17 pins can be specified as a normal CMOS output or N-ch open-drain output (V_{DD} tolerance **Note 1**/EV_{DD} tolerance **Note 2**) in 1-bit units using port output mode register 1 (POM1).

To use P13 and P14 as input pins, set them in the digital input mode or analog input mode by using port mode control register 1 (PMC1) (can be specified in 1-bit units).

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

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

- P10 to P12 and P05 to P07 pins ··· Input mode
- P13 and P14 pins ··· Analog input

Notes 1. 48-pin products: V_{DD} tolerance

2. 64-pin products: EV_{DD} tolerance

4.2.2 Port 2

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

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

To use P20/ANI0 to P21/ANI1 as digital input/output pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC). Use these pins starting from the upper bit.

To use P20/ANI0 to P21/ANI1 as analog input pins, set them in the analog input mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM2 register. Use these pins starting from the lower bit.

Table 4-2. Setting Functions of P20/ANI0 to P21/ANI1 Pins

ADPC Register	PM2 Register	ADS Register	P20/ANI0 to P21/ANI1 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P20/ANI0 to P21/ANI1 are set in the analog input mode when the reset signal is generated.

4.2.3 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 P32 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

This port can also be used for external interrupt request input, real-time clock correction clock output, segment output of the LCD controller/driver, and timer I/O.

Reset signal generation sets P30 to P32 to input mode.

4.2.4 Port 4

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 to P43 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).

Be sure to connect an external pull-up resistor to P40 when on-chip debugging is enabled (by using an option byte).

This port can also be used for segment output of the LCD controller/driver, external interrupt request input, data I/O for a flash memory programmer/debugger, timer I/O, and A/D converter analog input.

Reset signal generation sets P40, P42, and P43 to input mode, and sets P41 to analog input.

4.2.5 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 P54 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 the LCD controller/driver, and timer I/O.

Reset signal generation sets port 5 to input mode.

4.2.6 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 (EV_{DD} tolerance **Note**).

This port can also be used for segment output of the LCD controller/drive.

Reset signal generation sets port 6 to input mode.

Note 48-pin products: V_{DD} tolerance
64-pin products: EV_{DD} tolerance

4.2.7 Port 7

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

This port can also be used for key interrupt input and segment output of the LCD controller/driver.

Reset signal generation sets port 7 to input mode.

4.2.8 Port 12

P120 and P125 to P127 are 4-bit I/O ports with an output latch. These port pins can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

P121 to P124 are 4-bit input-only ports.

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

This port can also be used for A/D converter analog input, connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, external clock input for subsystem clock, segment output of the LCD controller/driver, connecting the capacitor for the LCD controller/driver, and voltage pin for driving the LCD.

Reset signal generation sets P120 to analog input, and sets P121 to P127 to input mode.

4.2.9 Port 13

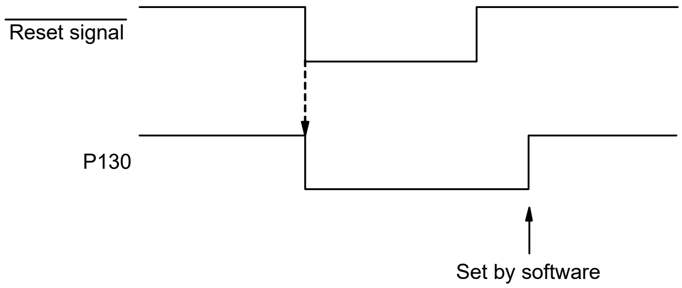
P130 is a 1-bit output-only port with an output latch.

P137 is a 1-bit input-only port.

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

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

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



4.2.10 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 P147 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 14 (PU14).

Digital input/output or analog input can be specified for the P142 to P145 pins using port mode control register 14 (PMC14).

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

Reset signal generation sets P140, P141, P146, and P147 to input mode, and sets P142 to P145 to analog input.

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 registers (PFSEG0 to PFSEG4)
- LCD input switch control register (ISCLCD)

Caution The undefined bits in each register vary by product and must be used with their initial value.

Table 4-3. PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (1/3)

Port		Bit name						64-pin	48-pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register		
Port 0	–	–	–	–	–	–	–	–	–
Port 1	0	PM10	P10	PU10	PIM10	POM10	–	√	√
	1	PM11	P11	PU11	PIM11	–	–	√	√
	2	PM12	P12	PU12	–	POM12	–	√	√
	3	PM13	P13	PU13	–	–	PMC13	√	√
	4	PM14	P14	PU14	–	–	PMC14	√	√
	5	PM15	P15	PU15	PIM15	POM15	–	√	√
	6	PM16	P16	PU16	PIM16	–	–	√	√
	7	PM17	P17	PU17	–	POM17	–	√	√
Port 2	0	PM20	P20	–	–	–	–	√	√
	1	PM21	P21	–	–	–	–	√	√
	2	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–

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

Port		Bit name						64-pin	48-pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register		
Port 3	0	PM30	P30	PU30	–	–	–	√	√
	1	PM31	P31	PU31	–	–	–	√	√
	2	PM32	P32	PU32	–	–	–	√	√
	3	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–
Port 4	0	PM40	P40	PU40	–	–	–	√	√
	1	PM41	P41	PU41	–	–	PMC41	√	√
	2	PM42	P42	PU42	–	–	–	√	–
	3	PM43	P43	PU43	–	–	–	√	–
	4	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–
Port 5	0	PM50	P50	PU50	–	–	–	√	√
	1	PM51	P51	PU51	–	–	–	√	–
	2	PM52	P52	PU52	–	–	–	√	–
	3	PM53	P53	PU53	–	–	–	√	–
	4	PM54	P54	PU54	–	–	–	√	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–
Port 6	0	PM60	P60	–	–	–	–	√	√
	1	PM61	P61	–	–	–	–	√	√
	2	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–
Port 7	0	PM70	P70	PU70	–	–	–	√	√
	1	PM71	P71	PU71	–	–	–	√	–
	2	PM72	P72	PU72	–	–	–	√	–
	3	PM73	P73	PU73	–	–	–	√	–
	4	PM74	P74	PU74	–	–	–	√	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–

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

Port		Bit name						64-pin	48-pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register		
Port 8	–	–	–	–	–	–	–	–	–
Port 9	–	–	–	–	–	–	–	–	–
Port 10	–	–	–	–	–	–	–	–	–
Port 11	–	–	–	–	–	–	–	–	–
Port 12	0	PM120	P120	PU120	–	–	PMC120	√	√
	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	–	–	–	–	√	√
Port 14	0	PM140	P140	PU140	–	–	–	√	√
	1	PM141	P141	PU141	–	–	–	√	√
	2	PM142	P142	PU142	–	–	PMC142	√	√
	3	PM143	P143	PU143	–	–	PMC143	√	√
	4	PM144	P144	PU144	–	–	PMC144	√	√
	5	PM145	P145	PU145	–	–	PMC145	√	–
	6	PM146	P146	PU146	–	–	–	√	–
	7	PM147	P147	PU147	–	–	–	√	–
Port 15	–	–	–	–	–	–	–	–	–

The format of each register is described below. The description here uses the 64-pin products as an example. For the registers mounted on others than 64-pin products, refer to **table 4-3**.

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 (64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	1	1	1	1	1	1	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	1	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26H	FFH	R/W
PM7	1	1	1	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM12	PM127	PM126	PM125	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	PM147	PM146	PM145	PM144	PM143	PM142	PM141	PM140	FFF2EH	FFH	R/W

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

Caution Be sure to set bits that are not 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 P13, P14, P20, P21, P41, P120, and P142 to P145 are set up as analog inputs of the A/D converter, when a port is read while in the input mode, 0 is always returned, not the pin level.

Figure 4-2. Format of Port Register (64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	0	0	0	0	0	0	P21	P20	FFF02H	00H (output latch)	R/W
P3	0	0	0	0	0	P32	P31	P30	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	P43	P42	P41	P40	FFF04H	00H (output latch)	R/W
P5	0	0	0	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	0	0	0	P74	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W
P12	P127	P126	P125	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W [Note 1]
P13	P137	0	0	0	0	0	0	P130	FFF0DH	Note 2	R/W [Note 1]
P14	P147	P146	P145	P144	P143	P142	P141	P140	FFF0EH	00H (output latch)	R/W

Pmn	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

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

2. P137 : Undefined
P130: 0 (output latch)

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

Remark m = 1 to 7, 12 to 14; n = 0 to 7

4.3.3 Pull-up resistor option registers (PUxx)

These registers specify whether the on-chip pull-up resistors are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode (PMmn = 1 and POMmn = 0) for the pins to which the use of an on-chip pull-up resistor has been specified in these registers. On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins and analog setting (PMC = 1, ADPC = 1), regardless of the settings of these registers.

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

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

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

Figure 4-3. Format of Pull-up Resistor Option Register (64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	PU32	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	PU43	PU42	PU41	PU40	F0034H	01H	R/W
PU5	0	0	0	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU7	0	0	0	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU12	PU127	PU126	PU125	0	0	0	0	PU120	F003CH	00H	R/W
PU14	PU147	PU146	PU145	PU144	PU143	PU142	PU141	PU140	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 1, 3 to 5, 7, 12, 14; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Cautions 1. For the pins used as LCD function pins (SEGxx, CAPL, CAPH, and V_{L3}), be sure to clear the corresponding PUmn bit of the PUm register to 0.

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

4.3.4 Port input mode register (PIM1

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

TTL input buffer can be selected during serial communication, etc 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 (64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM1	0	PIM16	PIM15	0	0	0	PIM11	PIM10	F0041H	00H	R/W

PIM1n	P1n pin input buffer selection (n = 0, 1, 5, 6)
0	Normal input buffer
1	TTL input buffer

Cautions 1. When using P10/SCK00/SEG28, P11/SI00/RxD0/TOOLRxD/SEG29, P15/SCK01/INTP1/SEG4, and P16/SI01/INTP2/SEG5 as LCD controller/driver function pins (segment output pins), setting the PIM1n bit to 1 is prohibited.

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

4.3.5 Port output mode register (POM1)

This register set the output mode of P10, P12, P15, P17 in 1-bit units.

N-ch open drain output (V_{DD} tolerance[Note 1]/EV_{DD} tolerance[Note 2]) mode can be selected during serial communication with an external device of the different potential.

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

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

Reset signal generation clears this register 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[Note 1]/EV_{DD} tolerance [Note 2]) mode is set.

Figure 4-5. Format of Port Input Mode Register (64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM1	POM17	0	POM15	0	0	POM12	0	POM10	F0051H	00H	R/W

POM1n	Pmn pin output mode selection (n = 0, 2, 5, 7)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance[Note 1]/EV_{DD} tolerance[Note 2]) mode

Notes 1. 48-pin products : V_{DD} tolerance

2. 64-pin products : EV_{DD} tolerance

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 P13, P14, P41, P120, and P142 to P145 digital I/O/analog input in 1-bit units.

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

Reset signal generation clears these registers to FFH.

Figure 4-6. Format of Port Mode Control Register (64-pin products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC1	1	1	1	PMC14	PMC13	1	1	1	F0061H	FFH	R/W
PMC4	1	1	1	1	1	1	PMC41	1	F0064H	FFH	R/W
PMC12	1	1	1	1	1	1	1	PMC120	F006CH	FFH	R/W
PMC14	1	1	PMC145	PMC144	PMC143	PMC142	1	1	F006EH	FFH	R/W

PMCmn	Pmn pin digital I/O/analog input selection (m = 1, 4, 12, 14; n = 0 to 5)
0	Digital I/O (alternate function other than analog input)
1	Analog input

Cautions 1. Select input mode by using port mode registers 1, 4, 12, 14 (PM1, PM4, PM12, PM14) for the ports which are set by the PMCxx register as analog input.

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

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

4.3.7 A/D port configuration register (ADPC)

This register switches the P20/ANI0, P21/AN21 pins to digital I/O of port or analog input of A/D converter.

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

Reset signal generation sets this register to 00H.

Figure 4-7. 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	0	ADPC1	ADPC0

ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching	
		ANI1/P21	ANI0/P20
0	0	A	A
0	1	D	D
1	0	D	A
1	1	A	A

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

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

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

4.3.8 Peripheral I/O redirection register (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	0	0	PIOR1	PIOR0

Bit	Function	64-pin		48-pin	
		Setting value		Setting value	
		0	1	0	1
PIOR1	PCLBUZ0	P140	P50	P140	P50
PIOR0	INTP1	P15	P53	P15	P10
	INTP2	P16	P54	P16	P11
	INTP6	P52	P140	–	–
	INTP7	P43	P141	–	–
	TI02/TO02	P17	P54	P17	P12

4.3.9 LCD port function registers 0 to 4 (PFSEG0 to PFSEG4)

These registers set whether to use pins P10 to P17, P30 to P32, P41 to P43, P50 to P54, P60, P61, P70 to P74, P120, and P140 to P147 as port pins (other than segment output pins) or segment output pins.

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

Reset signal generation sets these registers to FFH (PFSEG0 is set to F0H, and PFSEG4 is set to 7FH).

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

Figure 4-9. Format of LCD port function registers 0 to 4 (PFSEG0 to PFSEG4) (64-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: 7FH R/W

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

PFSEGxx (xx = 04 to 38)	Port (other than segment output)/segment outputs specification of Pmn pins (mn = 10 to 17, 30 to 32, 41 to 43, 50 to 54, 60, 61, 70-74, 120, 140 to 147)
0	Used the Pmn pin as port (other than segment output)
1	Used the Pmn pin as segment output

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

Bit name of PFSEG register	Corresponding SEGxx pins	Alternate port	64-pin	48-pin
PFSEG04	SEG4	P15	√	√
PFSEG05	SEG5	P16	√	√
PFSEG06	SEG6	P17	√	√
PFSEG07	SEG7	P50	√	√
PFSEG08	SEG8	P51	√	–
PFSEG09	SEG9	P52	√	–
PFSEG10	SEG10	P53	√	–
PFSEG11	SEG11	P54	√	–
PFSEG12	SEG12	P74	√	–
PFSEG13	SEG13	P73	√	–
PFSEG14	SEG14	P72	√	–
PFSEG15	SEG15	P71	√	–
PFSEG16	SEG16	P70	√	√
PFSEG17	SEG17	P32	√	√
PFSEG18	SEG18	P31	√	√
PFSEG19	SEG19	P30	√	√
PFSEG20	SEG20	P61	√	√
PFSEG21	SEG21	P60	√	√
PFSEG22	SEG22	P43	√	–
PFSEG23	SEG23	P42	√	–
PFSEG24	SEG24	P41	√	√
PFSEG25	SEG25	P120	√	√
PFSEG26	SEG26	P141	√	√
PFSEG27	SEG27	P140	√	√
PFSEG28	SEG28	P10	√	√
PFSEG29	SEG29	P11	√	√
PFSEG30	SEG30	P12	√	√
PFSEG31	SEG31	P13	√	√
PFSEG32	SEG32	P14	√	√
PFSEG33	SEG33	P142	√	√
PFSEG34	SEG34	P143	√	√
PFSEG35	SEG35	P144	√	√
PFSEG36	SEG36	P145	√	–
PFSEG37	SEG37	P146	√	–
PFSEG38	SEG38	P147	√	–

Remark √: Supported, –: Not supported

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. To use these pins as LCD function, input to the Schmitt trigger buffer must be disabled, in order to prevent through-currents from entering.

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

Reset signal generation sets these registers 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	Control of schmitt trigger buffer of V_{L3}/P125 pin
0	Makes digital input ineffective
1	Makes digital input effective

ISCCAP	Control of schmitt trigger buffer of CAPL/ P126 and CAPH/P127 pins
0	Makes digital input ineffective
1	Makes digital input effective

Cautions 1. If ISCVL3 bit = 0, set the corresponding port control registers as follows:
PU125 bit of PU12 register = 0, P125 bit of P12 register = 0

2. If ISCCAP bit = 0, set the corresponding port control 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

4.4 Port Function Operations

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

4.4.1 Writing to I/O port

(1) Output mode

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

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change. Therefore, byte data can be written to the ports used for both input and output.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

4.4.2 Reading from I/O port

(1) Output mode

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

(2) Input mode

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

4.4.3 Operations on I/O port

(1) Output mode

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

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change. Therefore, byte data can be written to the ports used for both input and output.
The data of the output latch is cleared when a reset signal is generated.

4.4.4 Connecting to external device with different potential (1.8 V, 2.5 V, 3 V)

It is possible to connect to an external device with a different potential (1.8 V, 2.5 V or 3 V) by changing EV_{DD} to accord with the power supply of the connected device. In products in which EV_{DD} cannot be specified independently, I/O connection with an external device operating on 1.8 V, 2.5 V or 3 V is still possible via the serial interface and general-purpose port by using port 1.

4.4.5 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers

It is possible to connect an external device operating on a different potential (1.8 V, 2.5 V or 3 V) by switching I/O buffers with the port input mode register (PIM1) and port output mode register (POM1).

When receiving input from an external device with a different potential (1.8 V, 2.5 V or 3 V), set the port input mode register 1 (PIM1) on a bit-by-bit basis to enable normal input (CMOS)/TTL input buffer switching.

When outputting data to an external device with a different potential (1.8 V, 2.5 V or 3 V), set the port output mode register 1 (POM1) on a bit-by-bit basis to enable normal output (CMOS)/N-ch open drain (V_{DD} tolerance **Note 1**/EV_{DD} tolerance **Note 2**) switching.

The connection of a serial interface is described in the following.

Notes **1.** For 48-pin products
2. For 64-pin products

(1) Setting procedure when using input pins of UART0, CSI00, and CSI01 functions for the TTL input buffer

In case of UART0:	P11
In case of CSI00:	P10, P11
In case of CSI01:	P15, P16

<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 PIM1 register to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, refer to the DC characteristics when the TTL input buffer is selected.

<3> Enable the operation of the serial array unit and set the mode to the UART/Simplified SPI (CSI **Note**) mode.

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

(2) Setting procedure when using output pins of UART0, CSI00, and CSI01 functions in N-ch open-drain output mode

In case of UART0:	P12
In case of CSI00:	P10, P12
In case of CSI01:	P15, P17

<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 POM1 register to 1 to set the N-ch open drain output (V_{DD} tolerance Note 1 /EV_{DD} tolerance Note 2) mode.

<5> Enable the operation of the serial array unit and set the mode to the UART/Simplified SPI (CSI) mode.

<6> Set the corresponding bit of the PM1 register to the output mode. At this time, the output data is high level, so the pin is in the Hi-Z state.

Notes **1.** For 48-pin products
2. For 64-pin products

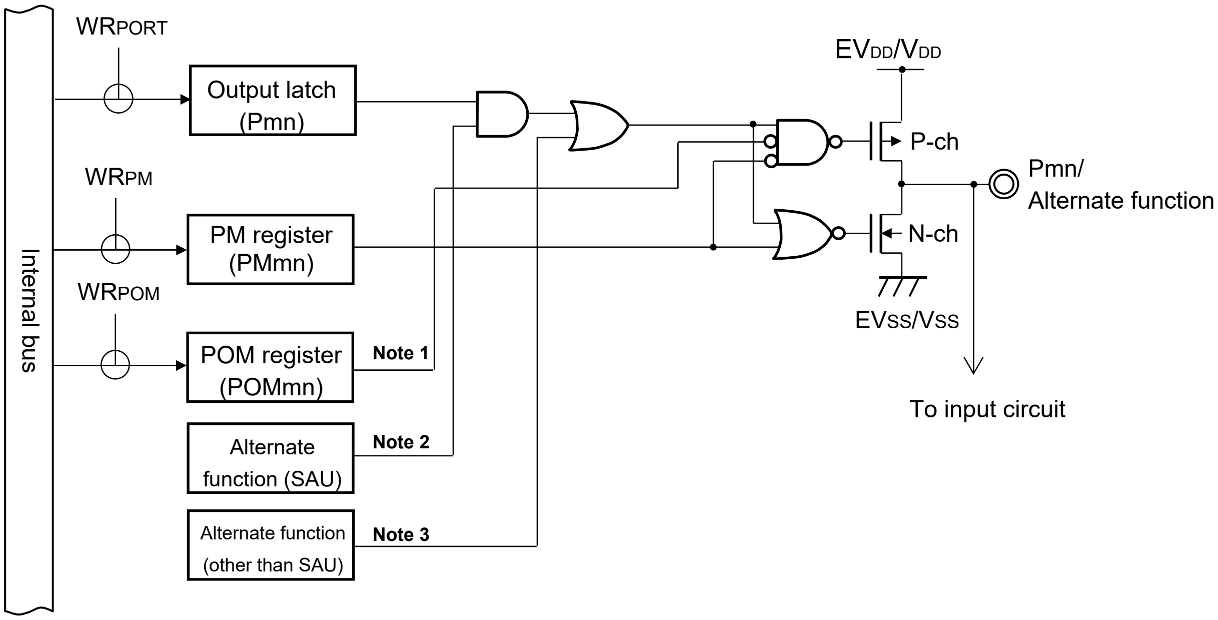
4.5 Register Settings When Using Alternate Function

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, RTC, 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 4-5.

Figure 4-11. Basic Configuration of Output Circuit for Pins



Notes **1.** When there is no POM register, this signal should be considered to be low level (0).
2. When there is no alternate function, this signal should be considered to be high level (1).
3. When there is no alternate function, this signal should be considered to be low level (0).

Remark m: Port number (m = 1 to 7, 12 to 14); n: Bit number (n = 0 to 7)

Table 4-5. 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.

4.5.3 Register setting examples for used port and alternate functions

Register setting examples for used port and alternate functions are shown in Table 4-6. The registers used to control the port functions should be set as shown in Table 4-6. See the following remark for legends used in Table 4-6.

Remark –: Not supported
×: don’t care
PIORx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch
PUxx: Pull-up resistor option register
PIMxx: Port input mode register
PFSEG xx: LCD port function register
ISCLCD: LCD input switch control register

Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (1/6)

	Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PUxx	PIMxx	PFSEGxx	ISCLCD
		Function Name	I/O									
	P10	P10	Input	×	×	—	1	×	0/1	0/1	0	—
			Output	×	0/1	—	0	0/1	×	×	0	—
<R>		SCK00	Input	×	×	—	1	×	0/1	0/1	0	—
			Output	×	0/1	—	0	1	×	×	0	—
		SEG28	Output	×	0	—	0	0	0	0	1	—
	P11	P11	Input	×	—	—	1	×	0/1	0/1	0	—
			Output	×	—	—	0	0/1	×	×	0	—
		SI00	Input	×	—	—	1	×	0/1	0/1	0	—
		RxD0	Input	×	—	—	1	×	0/1	0/1	0	—
		TOOLRxD	Input	×	—	—	1	×	0/1	0/1	0	—
		SEG29	Output	×	—	—	0	0	0	0	1	—
	P12	P12	Input	×	×	—	1	×	0/1	—	0	—
			Output	×	0/1	—	0	0/1	×	—	0	—
		SO00	Output	×	0/1	—	0	1	×	—	0	—
		TxD0	Output	×	0/1	—	0	1	×	—	0	—
		TOOLTxD	Output	×	0/1	—	0	1	×	—	0	—
		SEG30	Output	×	0	—	0	0	0	—	1	—
	P13	P13	Input	×	—	0	1	×	0/1	—	0	—
			Output	×	—	0	0	0/1	×	—	0	—
		ANI18	Input	×	—	1	1	×	×	—	—	—
		SEG31	Output	×	—	0	0	0	0	—	1	—
	P14	P14	Input	×	—	0	1	×	0/1	—	0	—
			Output	×	—	0	0	0/1	×	—	0	—
		ANI19	Input	×	—	1	1	×	×	—	—	—
		SEG32	Output	×	—	0	0	0	0	—	1	—
	P15	P15	Input	—	×	—	1	×	0/1	0/1	0	—
			Output	—	0/1	—	0	0/1	×	×	0	—
<R>		SCK01	Input	×	×	—	1	×	0/1	0/1	0	—
			Output	×	0/1	—	0	1	×	×	0	—
		INTP1	Input	PIOR0 = 0	×	—	1	×	0/1	0/1	0	—
		SEG4	Output	×	0	—	0	0	0	0	1	—
	P16	P16	Input	×	—	—	1	×	0/1	0/1	0	—
			Output	×	—	—	0	0/1	×	×	0	—
		SI01	Input	×	—	—	1	×	0/1	0/1	0	—
		INTP2	Input	PIOR0 = 0	—	—	1	×	0/1	0/1	0	—
		SEG5	Output	×	—	—	0	0	0	0	1	—
	P17	P17	Input	×	×	—	1	×	0/1	—	0	—
			Output	×	0/1	—	0	0/1	×	—	0	—
		SO01	Output	×	0/1	—	0	1	×	—	0	—
		TI02	Input	PIOR0 = 0	×	—	1	×	0/1	—	0	—
		TO02	Output	PIOR0 = 0	0/1	—	0	0	×	—	0	—
		SEG6	Output	×	0	—	0	0	0	—	1	—

Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (2/6)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PUxx	PIMxx	PFSEGxx	ISCLCD
	Function Name	I/O									
P20	P20	Input	×	—	—	1	×	—	—	—	—
		Output	×	—	—	0	0/1	—	—	—	—
	ANI0	Input	×	—	—	1	×	—	—	—	—
	AV_{REFP}	Input	×	—	—	1	×	—	—	—	—
P21	P21	Input	×	—	—	1	×	—	—	—	—
		Output	×	—	—	0	0/1	—	—	—	—
	ANI1	Input	×	—	—	1	×	—	—	—	—
	AV_{REFM}	Input	×	—	—	1	×	—	—	—	—
P30	P30	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	1	0/1	×	—	0	—
	TI01	Input	×	—	—	1	×	0/1	—	0	—
	TO01	Output	×	—	—	0	0	×	—	0	—
	SEG19	Output	×	—	—	0	0	0	—	1	—
P31	P31	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	INTP3	Input	×	—	—	1	×	0/1	—	0	—
	RTC1HZ	Output	×	—	—	0	0	×	—	0	—
	SEG18	Output	×	—	—	0	0	0	—	1	—
P32	P32	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	TI03	Input	×	—	—	1	×	0/1	—	0	—
	TO03	Output	×	—	—	0	0	×	—	0	—
	INTP4	Input	×	—	—	1	×	0/1	—	0	—
	SEG17	Output	×	—	—	0	0	0	—	1	—
P40	P40	Input	×	—	—	1	×	0/1	—	—	—
		Output	×	—	—	0	0/1	×	—	—	—
	TOOL0	I/O	×	—	—	×	×	×	—	—	—
P41	P41	Input	×	—	0	1	×	0/1	—	0	—
		Output	×	—	0	0	0/1	×	—	0	—
	ANI16	Input	×	—	1	1	×	×	—	—	—
	SEG24	Output	×	—	0	0	0	0	—	1	—
P42	P42	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	SEG23	Output	×	—	—	0	0	0	—	1	—
P43	P43	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	INTP7	Input	PIOR0 = 0	—	—	1	×	0/1	—	0	—
	SEG22	Output	×	—	—	0	0	0	—	1	—

Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (3/6)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PUxx	PIMxx	PFSEGxx	ISCLCD
	Function Name	I/O									
P50	P50	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	INTP5	Input	×	—	—	1	×	0/1	—	0	—
	SEG7	Output	×	—	—	0	0	0	—	1	—
	(PCLBUZ0)	Output	PIOR1 = 1	—	—	0	0	×	—	0	—
P51	P51	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	TI06	Input	×	—	—	1	×	0/1	—	0	—
	TO06	Output	×	—	—	0	0	×	—	0	—
	SEG8	Output	×	—	—	0	0	0	—	1	—
P52	P52	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	INTP6	Input	PIOR0 = 0	—	—	1	×	0/1	—	0	—
	SEG9	Output	×	—	—	0	0	0	—	1	—
P53	P53	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	TI07	Input	×	—	—	1	×	0/1	—	0	—
	TO07	Output	×	—	—	0	0	×	—	0	—
	SEG10	Output	×	—	—	0	0	0	—	1	—
	(INTP1)	Input	PIOR0 = 1	—	—	1	×	0/1	—	0	—
P54	P54	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	SEG11	Output	×	—	—	0	0	0	—	1	—
	(TI02)	Input	PIOR0 = 1	—	—	1	×	0/1	—	0	—
	(TO02)	Output	PIOR0 = 1	—	—	0	0	×	—	0	—
	(INTP2)	Input	PIOR0 = 1	—	—	1	×	0/1	—	0	—

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

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PUxx	PIMxx	PFSEGxx	ISCLCD
	Function Name	I/O									
P60	P60	Input	×	—	—	1	×	—	—	0	—
		Output	×	—	—	0	0/1	—	—	0	—
	SEG21	Output	×	—	—	0	0	—	—	1	—
P61	P61	Input	×	—	—	1	×	—	—	0	—
		Output	×	—	—	0	0/1	—	—	0	—
	SEG20	Output	×	—	—	0	0	—	—	1	—
P70	P70	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	KR0	Input	×	—	—	1	×	0/1	—	0	—
	SEG16	Output	×	—	—	0	0	0	—	1	—
P71	P71	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	KR1	Input	×	—	—	1	×	0/1	—	0	—
	SEG15	Output	×	—	—	0	0	0	—	1	—
P72	P72	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	KR2	Input	×	—	—	1	×	0/1	—	0	—
	SEG14	Output	×	—	—	0	0	0	—	1	—
P73	P73	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	KR3	Input	×	—	—	1	×	0/1	—	0	—
	SEG13	Output	×	—	—	0	0	0	—	1	—
P74	P74	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	SEG12	Output	×	—	—	0	0	0	—	1	—

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

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PUxx	PIMxx	PFSEGxx	ISCLCD
	Function Name	I/O									
P120	P120	Input	×	—	0	1	×	0/1	—	0	—
		Output	×	—	0	0	0/1	×	—	0	—
	ANI17	Input	×	—	1	1	×	×	—	—	—
	SEG25	Output	×	—	0	0	0	0	—	1	—
P121	P121	Input	×	—	—	—	×	—	—	—	—
		Output	×	—	—	—	0/1	—	—	—	—
	X1	Input	×	—	—	—	×	—	—	—	—
P122	P122	Input	×	—	—	—	×	—	—	—	—
		Output	×	—	—	—	0/1	—	—	—	—
	X2	Input	×	—	—	—	×	—	—	—	—
	EXCLK	Input	×	—	—	—	×	—	—	—	—
P123	P123	Input	×	—	—	—	×	—	—	—	—
		Output	×	—	—	—	0/1	—	—	—	—
	XT1	Input	×	—	—	—	×	—	—	—	—
P124	P124	Input	×	—	—	—	×	—	—	—	—
		Output	×	—	—	—	0/1	—	—	—	—
	XT2	Input	×	—	—	—	×	—	—	—	—
	EXCLKS	Input	×	—	—	—	×	—	—	—	—
P125	P125	Input	×	—	—	1	×	0/1	—	—	0
		Output	×	—	—	0	0/1	×	—	—	0
	V_{L3}	I/O	×	—	—	1	0	0	—	—	1
P126	P126	Input	×	—	—	1	×	0/1	—	—	0
		Output	×	—	—	0	0/1	×	—	—	0
	CAPL	I/O	×	—	—	1	0	0	—	—	1
P127	P127	Input	×	—	—	1	×	0/1	—	—	0
		Output	×	—	—	0	0/1	×	—	—	0
	CAPH	I/O	×	—	—	1	0	0	—	—	1

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

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	PUxx	PIMxx	PFSEGxx	ISCLCD
	Function Name	I/O									
P130	P130	Output	×	—	—	—	0/1	—	—	—	—
P137	P137	Input	×	—	—	—	×	—	—	—	—
	INTP0	Input	×	—	—	—	×	—	—	—	—
P140	P140	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	TO00	Output	×	—	—	0	0	×	—	0	—
	PCLBUZ0	Output	PIOR1 = 0	—	—	0	0	×	—	0	—
	SEG27	Output	×	—	—	0	0	0	—	1	—
	(INTP6)	Input	PIOR0 = 1	—	—	1	×	0/1	—	0	—
P141	P141	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	TI00	Input	×	—	—	1	×	0/1	—	0	—
	PCLBUZ1	Output	×	—	—	0	0	×	—	0	—
	SEG26	Output	×	—	—	0	0	0	—	1	—
	(INTP7)	Input	PIOR0 = 1	—	—	1	×	0/1	—	0	—
P142	P142	Input	×	—	0	1	×	0/1	—	0	—
		Output	×	—	0	0	0/1	×	—	0	—
	ANI20	Input	×	—	1	1	×	×	—	—	—
	SEG33	Output	×	—	0	0	0	0	—	1	—
P143	P143	Input	×	—	0	1	×	0/1	—	0	—
		Output	×	—	0	0	0/1	×	—	0	—
	ANI21	Input	×	—	1	1	×	×	—	—	—
	SEG34	Output	×	—	0	0	0	0	—	1	—
P144	P144	Input	×	—	0	1	×	0/1	—	0	—
		Output	×	—	0	0	0/1	×	—	0	—
	ANI22	Input	×	—	1	1	×	×	—	—	—
	SEG35	Output	×	—	0	0	0	0	—	1	—
P145	P145	Input	×	—	0	1	×	0/1	—	0	—
		Output	×	—	0	0	0/1	×	—	0	—
	ANI23	Input	×	—	1	1	×	×	—	—	—
	SEG36	Output	×	—	0	0	0	0	—	1	—
P146	P146	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	SEG37	Output	×	—	—	0	0	0	—	1	—
P147	P147	Input	×	—	—	1	×	0/1	—	0	—
		Output	×	—	—	0	0/1	×	—	0	—
	SEG38	Output	×	—	—	0	0	0	—	1	—

4.6 Cautions When Using Port Function

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

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

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

<Example> When 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 R7F0C001G/L, R7F0C002G/L.

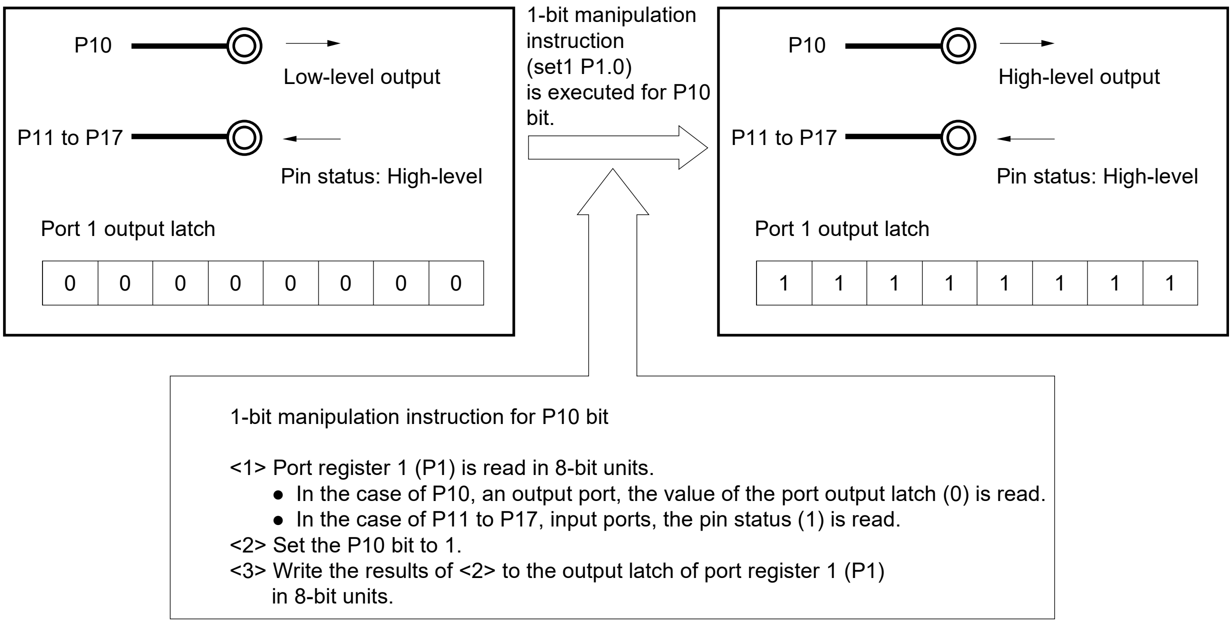
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

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 4-12. Bit Manipulation Instruction (P10)



1-bit manipulation instruction for P10 bit

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

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

4.6.2 Notes on specifying the pin settings

For an output pin to which multiple alternate functions are assigned, the output of the unused alternate function must be set to its initial state so as to prevent conflicting outputs. This also applies to the functions assigned by using the peripheral I/O redirection register (PIOR). For details about the alternate output function, see **4.5 Register Settings When Using Alternate Function**.

No specific setting is required for input pins because the output function 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 and X2 pins.
Oscillation can be stopped by executing the STOP instruction or setting of the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

<2> High-speed on-chip oscillator

The frequency at which to oscillate can be selected from among f_{IH} = 24, 16, 12, 8, 4, or 1 MHz (typ.) by using the option byte (000C2H). After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock. Oscillation can be stopped by executing the STOP instruction or setting the HIOSTOP bit (bit 0 of the CSC register).
The frequency specified by using an option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 5-9 Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV)**.
The frequencies that can be specified for the high-speed on-chip oscillator by using the option byte and the high-speed on-chip oscillator frequency select register (HOCODIV) are shown below.

Power Supply Voltage	Oscillation Frequency (MHz)								
	1	2	3	4	6	8	12	16	24
2.7 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	√	√
2.4 V ≤ V_{DD} < 2.7 V	√	√	√	√	√	√	√	√	–
1.8 V ≤ V_{DD} < 2.4 V	√	√	√	√	√	√	–	–	–
1.6 V ≤ V_{DD} < 1.8 V	√	√	–	√	–	–	–	–	–

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

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

(3) Low-speed on-chip oscillator clock

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

- Watchdog timer
- Real-time clock
- 12-bit interval timer
- LCD driver/controller

This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the subsystem clock supply mode control register, 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 operation clock when the fixed-cycle interrupt function is used.

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{EX}: External main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXS}: External subsystem clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency

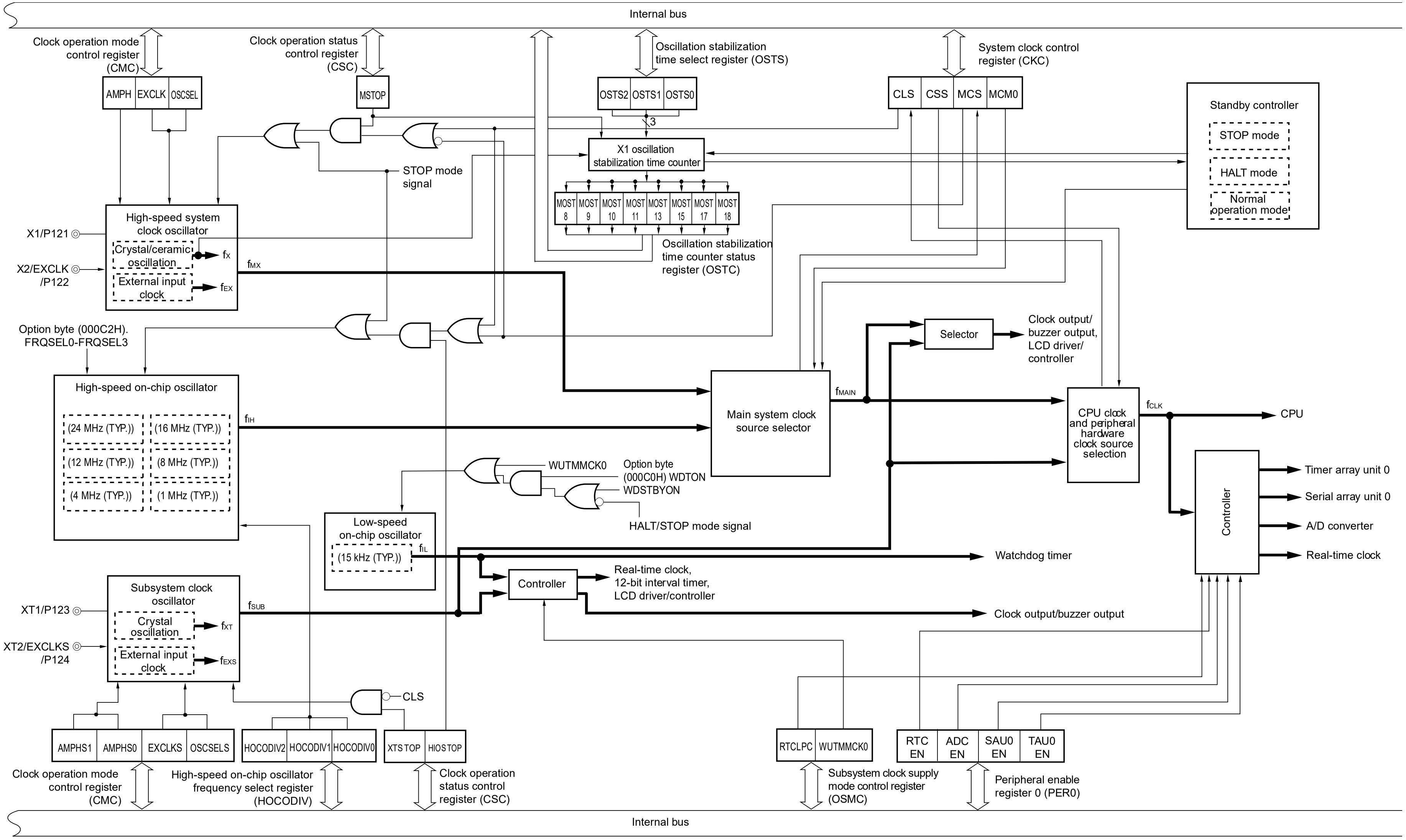
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

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

Figure 5-1. Block Diagram of Clock Generator



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

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

5.3 Registers Controlling Clock Generator

The following nine 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 register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)

5.3.1 Clock operation mode control register (CMC)

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

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

Reset signal generation clears this register to 00H.

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

Address: FFFA0H After reset: 00H R/W

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

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

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

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

AMPH	Control of X1 clock oscillation frequency
0	1 MHz $\leq f_X \leq$ 10 MHz
1	10 MHz $< f_X \leq$ 20 MHz

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

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

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

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

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

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

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

Cautions 7. The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **Before using the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) as the mode of the XT1 oscillator, evaluate the resonators.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential as V_{SS} as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

Remark f_X: X1 clock frequency

5.3.2 System clock control register (CKC)

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

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

Reset signal generation sets this register to 00H.

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

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

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

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

CSS	Selection of CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1 [Note 2]	Subsystem clock (f_{SUB})

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

MCM0 [Note 2]	Main system clock (f_{MAIN}) operation control
0	Selects the high-speed on-chip oscillator clock (f_{IH}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

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

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

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

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

3. If the subsystem clock is used as the peripheral hardware clock, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 27 ELECTRICAL SPECIFICATIONS.

Remarks 1. f_{IH}: High-speed on-chip oscillator clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency

2. ×: don’t care

5.3.3 Clock operation status control register (CSC)

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

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

Reset signal generation sets this register to C0H.

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

Address: FFFA1H After reset: C0H R/W

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

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

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

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

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

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

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

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

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

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

Table 5-2. Stopping Clock Method

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

5.3.4 Oscillation stabilization time counter status register (OSTC)

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

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

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

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

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

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

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

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

Address: FFFA2H After reset: 00H R

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

MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									f_X = 10 MHz	f_X = 20 MHz
0	0	0	0	0	0	0	0	$2^8/f_X$ max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	$2^8/f_X$ min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	$2^9/f_X$ min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	$2^{10}/f_X$ min.	102.4 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	$2^{11}/f_X$ min.	204.8 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	$2^{13}/f_X$ min.	819.2 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	$2^{15}/f_X$ min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	$2^{17}/f_X$ min.	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	$2^{18}/f_X$ min.	26.21 ms min.	13.11 ms min.

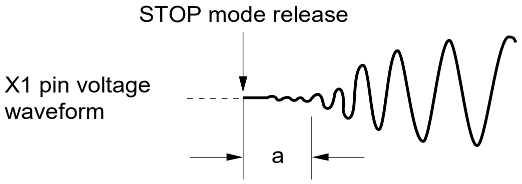
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.

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

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

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

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



Remark f_X: X1 clock oscillation frequency

5.3.5 Oscillation stabilization time select register (OSTS)

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

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

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

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

Reset signal generation sets the OSTS register to 07H.

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

Address: FFFA3H After reset: 07H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$2^{8}/f_X$	25.6 μs	12.8 μs
0	0	1	$2^{9}/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_X$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_X$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.21 ms	13.11 ms

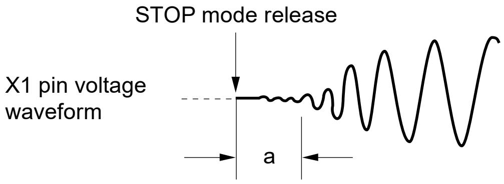
Cautions 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

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

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

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

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



Remark f_X: X1 clock oscillation frequency

5.3.6 Peripheral enable register 0 (PER0)

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

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

- Real-time clock, 12-bit interval timer
- A/D converter
- Serial array unit 0
- Timer array unit 0
- LCD driver/controller

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

Reset signal generation clears these registers to 00H.

Figure 5-7. Format of Peripheral Enable Register 0 (PER0) (1/2)

Address: F00F0H After reset: 00H R/W

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

RTCEN	Real-time clock (RTC) and 12-bit interval timer	LCD driver/controller and clock output/buzzer output controller	
		When subsystem clock (f_{SUB}) is selected.	When subsystem clock (f_{SUB}) is not selected.
0	Stops input clock supply. ● SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. ● The real-time clock (RTC) and 12-bit interval timer are in the reset status.	Stops input clock and subsystem clock supply. ● SFR used by the LCD driver/controller and clock output/buzzer output can be read and written.	Enables input clock and main system clock supply. ● SFR used by the LCD driver/controller and clock output/buzzer output can be read and written.
1	Enables input clock supply. ● SFR used by the real-time clock (RTC) and 12-bit interval timer can be read and written.	Enables input clock and subsystem clock supply. ● SFR used by the LCD driver/controller and clock output/buzzer output can be read and written.	

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

Caution Be sure to clear the following bits 1, 3, and 6 to 0.

Figure 5-7. Format of Peripheral Enable Register 0 (PER0) (2/2)

Address: F00F0H After reset: 00H R/W

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

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

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

Caution Be sure to clear the following bits 1, 3, 4 and 6 to 0.

5.3.7 Subsystem clock supply mode control register (OSMC)

This register is used to reduce power consumption by stopping as many unnecessary clock functions.

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions, except the real-time clock, 12-bit interval timer, and LCD driver/controller, is stopped in STOP mode and HALT mode while subsystem clock is selected as CPU clock. Set bit 7 (RTCEN) of peripheral enable registers 0 (PER0) to 1 before this setting.

In addition, the OSMC register can be used to select the operation clock of the real-time clock, 12-bit interval timer, 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 5-8. 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 HALT mode while subsystem clock is selected as CPU clock
0	Enables supply of subsystem clock to peripheral functions (See **Table 19-1** for peripheral functions whose operations are enabled.)
1	Stops supply of subsystem clock to peripheral functions other than real-time clock, 12-bit interval timer, and LCD driver/controller.

WUTMMCK0 Note	Selection of operation clock for real-time clock, 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.

Cautions 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, 12-bit interval timer, and LCD driver/controller are all stopped.

These are stopped as follows:

Real-time clock:	**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.**

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

Remark

RTCE:	Bit 7 of real-time clock control register 0 (RTCC0)
RINTE:	Bit 15 of the 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 FRQSEL3 bit of the option byte (000C2H).

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

Reset signal generation clears this register to default value (undefined).

Figure 5-9. Format of High-speed On-chip Oscillator frequency select register (HOCODIV)

Address: F00A8H After reset: undefined R/W

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

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

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

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE2			
0	0	LV (low-voltage main) mode	1 to 4 MHz	1.6 to 5.5 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 5.5 V
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 5.5 V
			1 to 32 MHz	2.7 to 5.5 V

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

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

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

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.

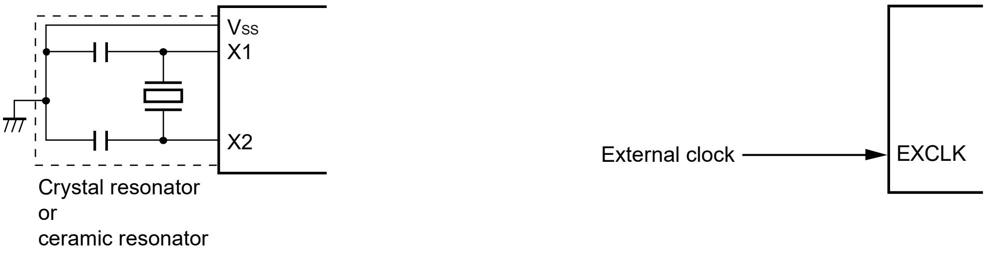
- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

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 (64-pin products)**.

Figure 5-10 shows an example of the external circuit of the X1 oscillator.

Figure 5-10. Example of External Circuit of X1 Oscillator



(a) Crystal or ceramic oscillation

(b) External clock

Cautions are listed on the next page.

5.4.2 XT1 oscillator

The XT1 oscillator oscillates with a crystal resonator (32.768 kHz (TYP.)) connected to the XT1 and XT2 pins.

To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

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

To use the XT1 oscillator, set bits 5 and 4 (EXCLKS, OSCSELS) of the clock operation mode control register (CMC) as follows.

- Crystal oscillation: EXCLKS, OSCSELS = 0, 1
- External clock input: EXCLKS, OSCSELS = 1, 1

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

When the pins are not used as input port pins, either, see **Table 2-3 Connection of Unused Pins (64-pin products)**.

Figure 5-11 shows an example of the external circuit of the XT1 oscillator.

Figure 5-11. 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 5-10 and 5-11 to avoid an adverse effect from wiring capacitance.**

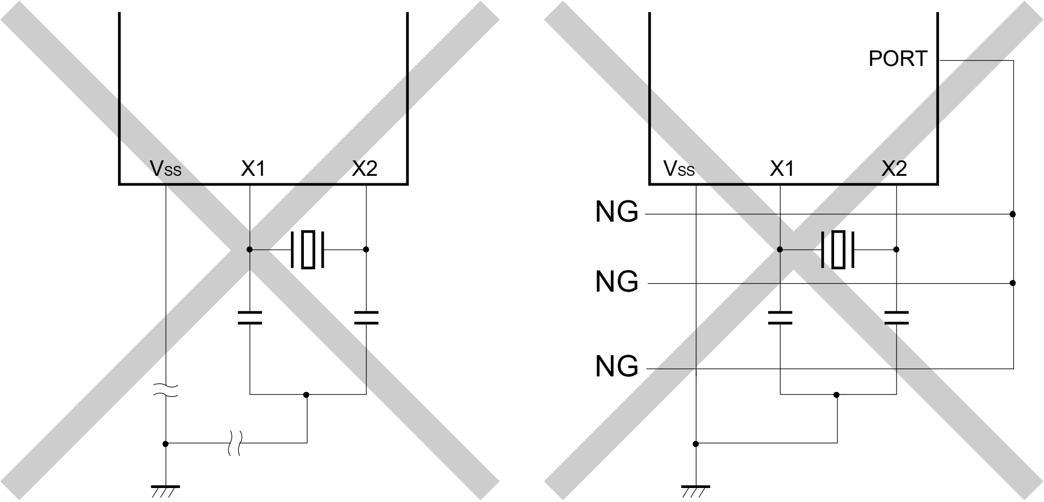
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground 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.**
- **Before using the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) as the mode of the XT1 oscillator, evaluate the resonators.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential as V_{SS} as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

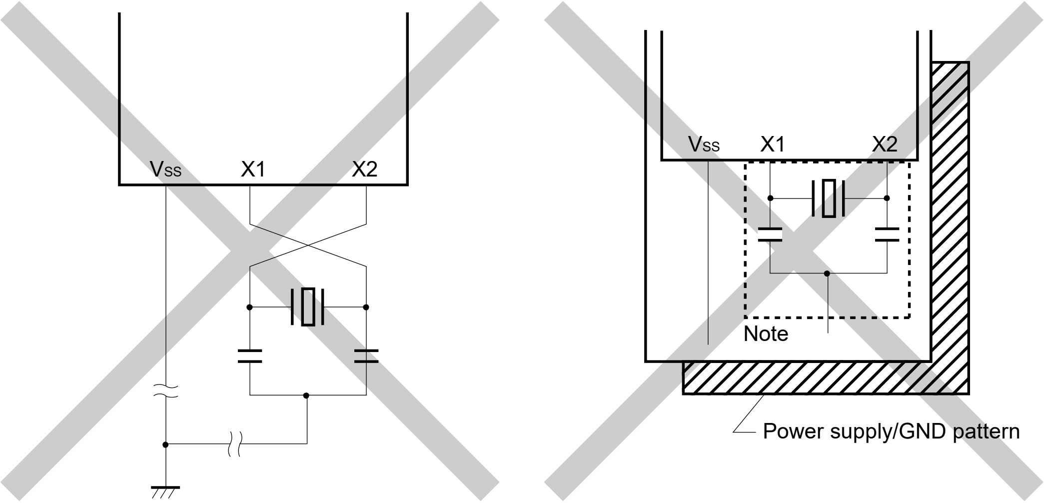
Figure 5-12 shows examples of incorrect resonator connection.

Figure 5-12. Examples of Incorrect Resonator Connection (1/2)



(a) Too long wiring

(b) Crossed signal line



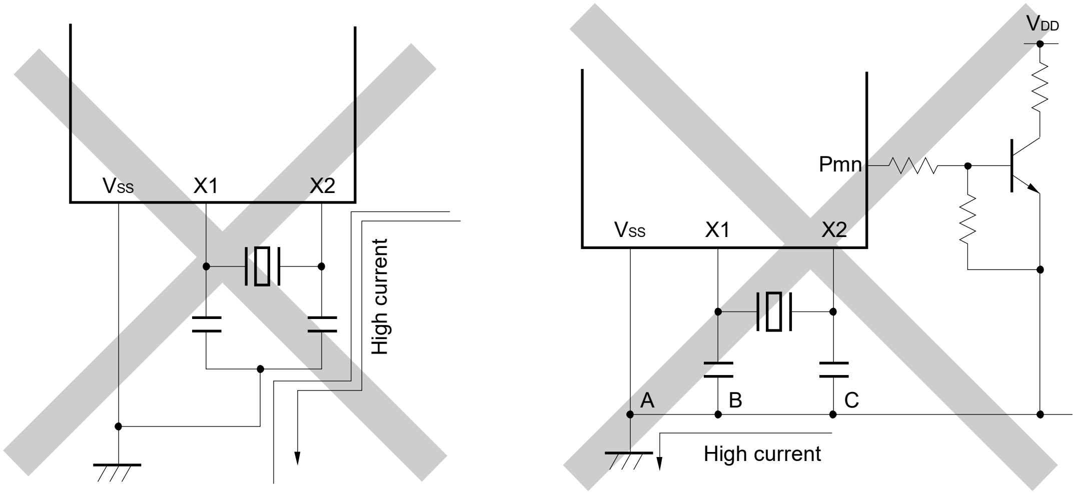
(c) The X1 and X2 signal line wires cross.

(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.

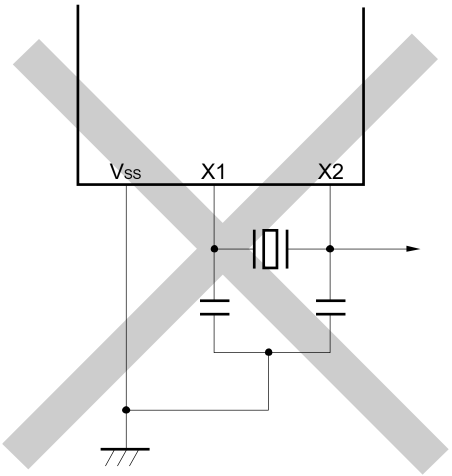
Figure 5-12. Examples of Incorrect Resonator Connection (2/2)



(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 R7F0C001G/L, R7F0C002G/L. The frequency can be selected from among 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using the option byte (000C2H). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC). The high-speed on-chip oscillator automatically starts oscillating after reset release.

5.4.4 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the R7F0C001G/L, R7F0C002G/L.

The low-speed on-chip oscillator clock is used only as a clock of the watchdog timer, real-time clock, 12-bit interval timer, and LCD driver/controller. The low-speed on-chip oscillation 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.

5.5 Clock Generator Operation

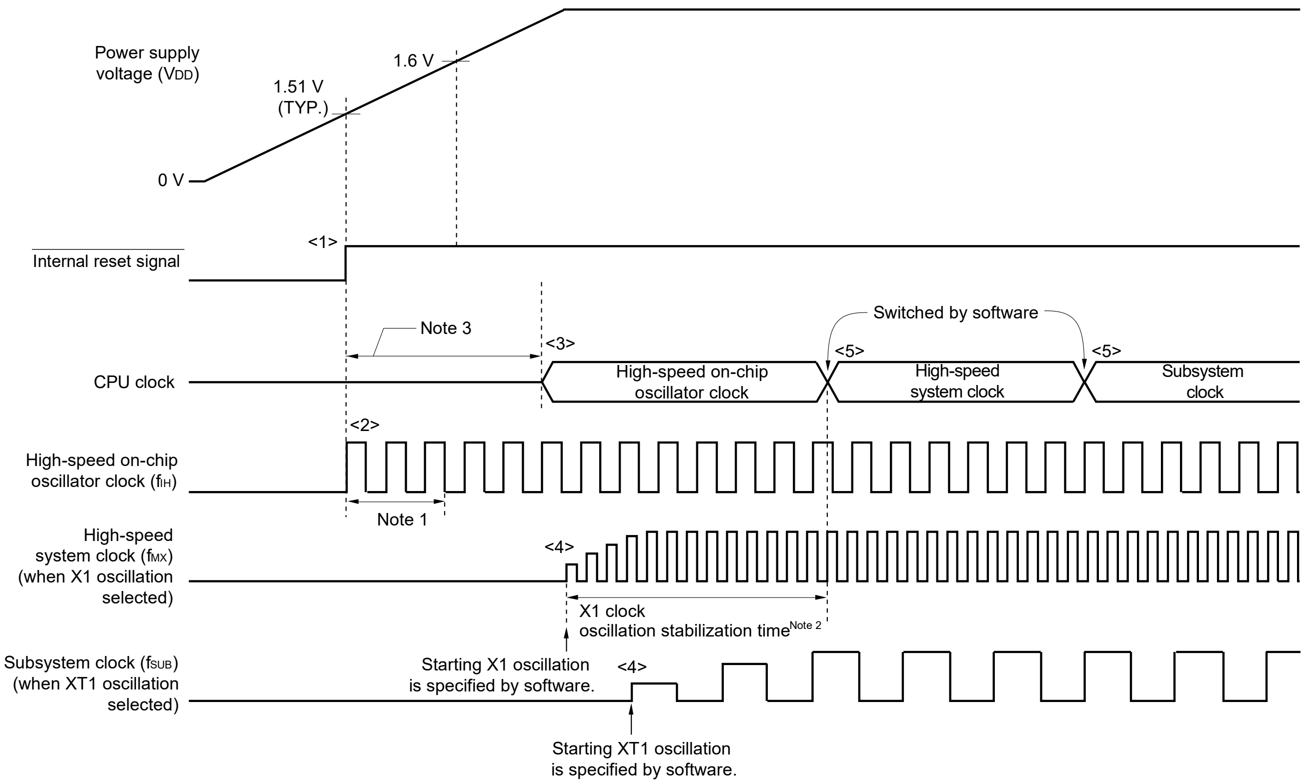
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 5-1**).

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

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the R7F0C001G/L, R7F0C002G/L.

When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-13.

Figure 5-13. 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.

<2> When the power supply voltage exceeds 1.51 V (TYP.), the reset is released and the high-speed on-chip oscillator automatically starts oscillation.

<3> The CPU starts operation on the high-speed on-chip oscillator clock after a reset processing such as waiting for the voltage of the power supply or regulator to stabilize has been performed after reset release.

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

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

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

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

3. For the reset processing time, see **CHAPTER 18 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 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz by using FRQSEL0 to FRQSEL3 of the option byte (000C2H).

[Option byte setting]
Address: 000C2H

	7	6	5	4	3	2	1	0
Option byte (000C2H)	CMODE1 0/1	CMODE0 0/1	1	0	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 5.5 V@1 MHz to 4 MHz
1	0	LS (low speed main) mode	V_{DD} = 1.8 V to 5.5 V@1 MHz to 8 MHz
1	1	HS (high speed main) mode	V_{DD} = 2.4 V to 5.5 V@1 MHz to 16 MHz V_{DD} = 2.7 V to 5.5 V@1 MHz to 24 MHz
Other than above		Setting prohibited	

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

[High-speed on-chip oscillator frequency select register (HOCODIV) setting]
Address: F00A8H

	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	
			FRQSEL3 Bit is 0	FRQSEL3 Bit is 1
0	0	0	24 MHz	Setting prohibited
0	0	1	12 MHz	16 MHz
0	1	0	6 MHz	8 MHz
0	1	1	3 MHz	4 MHz
1	0	0	Setting prohibited	2 MHz
1	0	1	Setting prohibited	1 MHz
Other than above			Setting prohibited	

5.6.2 Example of setting X1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the X1 oscillation clock, set the oscillator and start oscillation by using the oscillation stabilization time select register (OSTS), clock operation mode control register (CMC) 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 f_X > 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 102.4 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 102.4 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 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	CMODE2			
0	0	LV (low-voltage main) mode	1 to 4 MHz	1.6 to 5.5 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 5.5 V
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 5.5 V
			1 to 24 MHz	2.7 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> To run only the real-time clock, 12-bit interval timer, and LCD driver/controller on the subsystem clock (ultra-low current consumption) when in the STOP mode or HALT mode during CPU operation on the subsystem clock, set the RTCLPC bit to 1.

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

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

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

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

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

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

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

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

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

5.6.4 CPU clock status transition diagram

Figure 5-14 shows the CPU clock status transition diagram of this product.

Figure 5-14. CPU Clock Status Transition Diagram

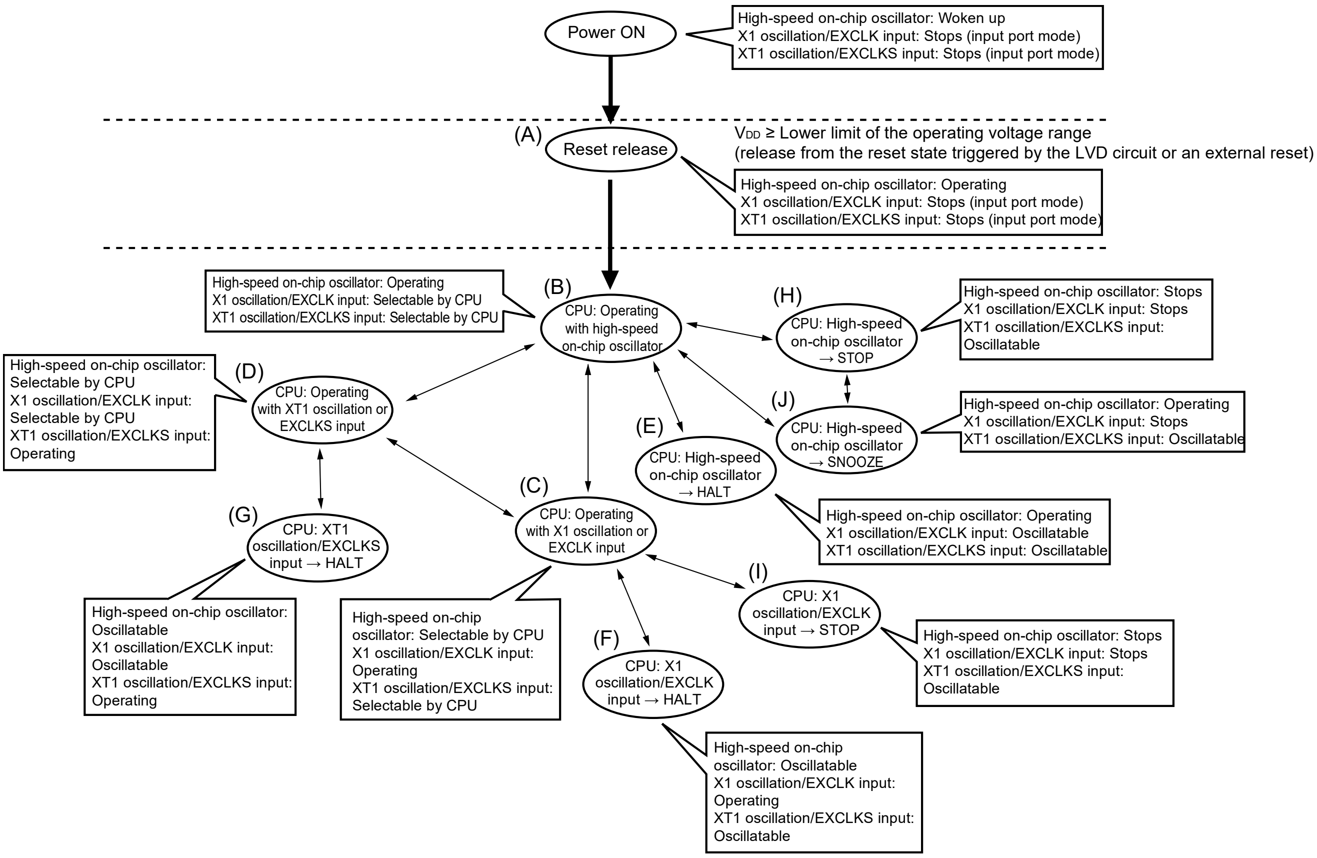


Table 5-3 shows transition of the CPU clock and examples of setting the SFR registers.

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

(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 [Note1]			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(A) → (B) → (C) (X1 clock: 1 MHz ≤ fx ≤ 10 MHz)	0	1	0	**Note 2**	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	**Note 2**	0	Must be checked	1
(A) → (B) → (C) (external main clock)	1	1	×	**Note 2**	0	Must not be checked	1

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

2. Set the oscillation stabilization time as follows.

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

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 27 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.

Remarks 1. ×: don't care

2. (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-14.

Table 5-3. 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	Must not be checked	1

Unnecessary if these registers are already set (EXCLK, OSCSEL, AMPH)

Unnecessary if the CPU is operating with the high-speed system clock (OSTS to CKC)

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

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 27 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	XTSTOP		CSS
(B) → (D) (XT1 clock)	0	1	0	Necessary	1
(B) → (D) (external sub clock)	1	1	0	Necessary	1

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.

Remarks 1. ×: don't care

2. (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-14.

Table 5-3. 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	18 μs to 65 μs	0

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

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

(Setting sequence of SFR registers) →

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

Unnecessary if the CPU is operating with the subsystem clock

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		CSS
(D) → (B)	0	18 μs to 65 μs	0

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

Remarks 1. (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-15.

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

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

(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
		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	Must not be checked	0

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

Note Set the oscillation stabilization time as follows.

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

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 27 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 Table 5-3 correspond to (A) to (J) in Figure 5-14.

Table 5-3. 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 cannot operate 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 **11.8 SNOOZE Mode Function**, **12.5.7 SNOOZE mode function** and **12.6.3 SNOOZE mode function**.

Remark (A) to (J) in Table 5-3 correspond to (A) to (J) in Figure 5-14.

5.6.5 Condition before changing CPU clock and processing after changing CPU clock

Condition before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 5-4. Changing CPU Clock (1/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
High-speed on-chip oscillator clock	X1 clock	Stabilization of X1 oscillation ● OSCSEL = 1, EXCLK = 0, MSTOP = 0 ● After elapse of oscillation stabilization time	After checking that the CPU clock is switched to the clock after change, operating current can be reduced by stopping high-speed on-chip oscillator (HIOSTOP = 1).
	External main system clock	Enabling input of external clock from the EXCLK pin ● OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	XT1 clock	Stabilization of XT1 oscillation ● OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 ● After elapse of oscillation stabilization time	
	External subsystem clock	Enabling input of external clock from the EXCLKS pin ● OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
X1 clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator ● HIOSTOP = 0 ● After elapse of oscillation accuracy stabilization time	After checking that the CPU clock is switched to the clock after change, X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	XT1 clock	Stabilization of XT1 oscillation ● OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 ● After elapse of oscillation stabilization time	After checking that the CPU clock is switched to the clock after change, X1 oscillation can be stopped (MSTOP = 1).
	External subsystem clock	Enabling input of external clock from the EXCLKS pin ● OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	After checking that the CPU clock is switched to the clock after change, X1 oscillation can be stopped (MSTOP = 1).
External main system clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator ● HIOSTOP = 0 ● After elapse of oscillation accuracy stabilization time	External main system clock input can be disabled (MSTOP = 1).
	X1 clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	XT1 clock	Stabilization of XT1 oscillation ● OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 ● After elapse of oscillation stabilization time	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).

Table 5-4. Changing CPU Clock (2/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
XT1 clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as main system clock ● HIOSTOP = 0, MCS = 0	XT1 oscillation can be stopped (XTSTOP = 1)
	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	
	High-speed on-chip oscillator clock	Transition not possible	–
External subsystem clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as main system clock ● HIOSTOP = 0, MCS = 0	External subsystem clock input can be disabled (XTSTOP = 1).
	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	–

5.6.6 Time required for switchover of CPU clock and system clock

By setting bits 4 and 6 (MCM0, CSS) of the system clock control register (CKC), the CPU clock can be switched (between the main system clock and the subsystem clock), and main system clock can be switched (between the high-speed on-chip oscillator clock and the high-speed system clock).

The actual switchover operation is not performed immediately after rewriting to the CKC register; operation continues on the pre-switchover clock for several clocks (see Table 5-5 to Table 5-7).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 7 (CLS) of the CKC register. Whether the main system clock is operating on the high-speed system clock or high-speed on-chip oscillator clock can be ascertained using bit 5 (MCS) of the CKC register.

When the CPU clock is switched, the peripheral hardware clock is also switched.

Table 5-5. Maximum Time Required for System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{IH}	↔	f_{MX}	See **Table 5-6**
f_{MAIN}	↔	f_{SUB}	See **Table 5-7**

Table 5-6. Maximum Number of Clocks Required for f_{IH} ↔ f_{MX}

Set Value Before Switchover		Set Value After Switchover	
MCM0		MCM0	
		0 ($f_{MAIN} = f_{IH}$)	1 ($f_{MAIN} = f_{MX}$)
0 ($f_{MAIN} = f_{IH}$)	$f_{MX} \geq f_{IH}$		2 clock
	$f_{MX} < f_{IH}$		$2f_{IH}/f_{MX}$ clock
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} \geq f_{IH}$	$2f_{MX}/f_{IH}$ clock	
	$f_{MX} < f_{IH}$	2 clock	

Table 5-7. Maximum Number of Clocks Required for f_{MAIN} ↔ f_{SUB}

Set Value Before Switchover	Set Value After Switchover	
CSS	CSS	
	0 ($f_{CLK} = f_{MAIN}$)	1 ($f_{CLK} = f_{SUB}$)
0 ($f_{CLK} = f_{MAIN}$)		1 + $2f_{MAIN}/f_{SUB}$ clock
1 ($f_{CLK} = f_{SUB}$)	3 clock	

Remarks 1. The number of clocks listed in Table 5-6 to Table 5-7 is the number of CPU clocks before switchover.

2. Calculate the number of clocks in Table 5-6 to Table 5-7 by removing the decimal portion.

Example When switching the main system clock from the high-speed system clock to the high-speed on-chip oscillator clock (@ oscillation with f_{IH} = 8 MHz, f_{MX} = 10 MHz)

$2f_{MX}/f_{IH} = 2\ (10/8) = 2.5 \rightarrow 3$ clocks

5.6.7 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 5-8. 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	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
External main system clock		
XT1 clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock.)	XTSTOP = 1
External subsystem clock		

5.7 Resonator and Oscillator Constants

For the resonators for which operation has been verified and their oscillator constants, see the target product page on the Renesas Web site.

Cautions 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.

2. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the R7F0C001G/L, R7F0C002G/L so that the internal operation conditions are within the specifications of the DC and AC characteristics.

Figure 5-15. External Oscillation Circuit Example

(a) X1 oscillation **(b) XT1 oscillation**

(1) X1 oscillation:

As of October, 2013

Manufacturer	Resonator	Part Number	SMD/ Lead	Frequency (MHz)	Flash operation mode Note 1	Recommended Circuit Constants Note 2 (reference)			Oscillation Voltage Range (V)	
						C1 (pF)	C2 (pF)	Rd (kΩ)	MIN.	MAX.
Murata Manufacturing Co., Ltd.	Ceramic resonator	CSTCC2M00G56-R0	SMD	2.0	LV	(47)	(47)	0	1.6	5.5
		CSTCR4M00G55-R0	SMD	4.0		(39)	(39)	0		
		CSTLS4M00G53-B0	Lead			(15)	(15)	0		
		CSTCC2M00G56-R0	SMD	2.0	LS	(47)	(47)	0	1.8	5.5
		CSTCR4M00G55-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	5.5
		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	5.5
		CSTLS20M0X51-B0	Lead			(5)	(5)	0		
Nihon Dempa Kogyo Co., Ltd.	Crystal resonator	NX8045GB Note 3	SMD	8.0	**Note 3**					
		NX5032GA Note 3	SMD	16.0						
		NX3225HA Note 3	SMD	20.0	HS	3	3	0	2.4	5.5

Notes 1. Set the flash operation mode by using CMODE1 and CMODE0 bits of the option byte (000C2H).

2. Values in parentheses in the C1 and C2 columns indicate an internal capacitance.

3. When using this resonator, for details about the matching, contact Nihon Dempa Kogyo Co., Ltd (http://www.ndk.com/en).

Remark Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: $2.7\ V \le V_{DD} \le 5.5\ V$@1 MHz to 24 MHz
$2.4\ V \le V_{DD} \le 5.5\ V$@1 MHz to 16 MHz
LS (low-speed main) mode: $1.8\ V \le V_{DD} \le 5.5\ V$@1 MHz to 8 MHz
LV (low-voltage main) mode: $1.6\ V \le V_{DD} \le 5.5\ V$@1 MHz to 4 MHz

(2) XT1 oscillation: Crystal resonator

As of October, 2013

Manufacturer	Part Number	SMD/ Lead	Frequency (kHz)	Load Capacitance CL (pF)	XT1 oscillation mode Note 1	Recommended Circuit Constants			Oscillation Voltage Range (V)	
						C3 (pF)	C4 (pF)	Rd (kΩ)	MIN.	MAX
Nihon Dempa Kogyo Co., Ltd.	NX3215SA Note 2	SMD	32.768	6	Normal oscillation	7	7	0	1.6	5.5
					Low power consumption oscillation	6	7			
					Ultra-low power consumption oscillation	6	6			
	NX2012SA Note 2	SMD	32.768	6	Normal oscillation	**Note 2**				
					Low power consumption oscillation					
					Ultra-low power consumption oscillation					

Notes **1.** Set the XT1 oscillation mode by using AMPHS0 and AMPHS1 bits of the clock operation mode control register (CMC).

2. When using this resonator, for details about the matching, contact Nihon Dempa Kogyo Co., Ltd (http://www.ndk.com/en).

CHAPTER 6 TIMER ARRAY UNIT

The number of units or channels of the timer array unit differs, depending on the product.

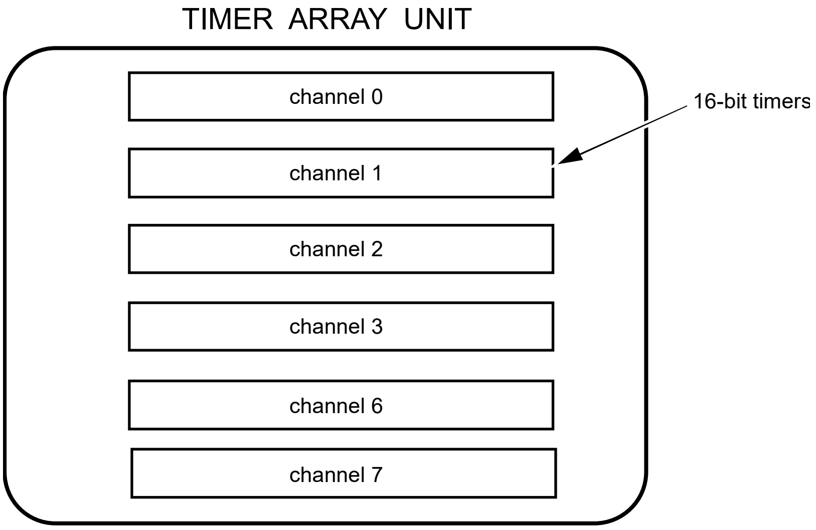
Channels	48-pin	64-pin
Channel 0	√	√
Channel 1	√	√
Channel 2	√	√
Channel 3	√	√
Channel 4	√	√
Channel 5	–	√
Channel 6	–	√
Channel 7	√	√

Cautions 1. The presence or absence of timer I/O pins depends on the product. See Table 6-2 Timer I/O Pins Provided in Each Product for details.

2. Most of the following descriptions in this chapter use the 64-pin products as an example.

The timer array unit has six 16-bit timers.

Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more “channels” can be used to create a high-accuracy timer.



For details about each function, see the table below.

Independent channel operation function	Simultaneous channel operation function
● Interval timer (→ refer to **6.8.1**) ● Square wave output (→ refer to **6.8.1**) ● External event counter (→ refer to **6.8.2**) ● Divider Note (→ refer to **6.8.3**) ● Input pulse interval measurement (→ refer to **6.8.4**) ● Measurement of high-/low-level width of input signal (→ refer to **6.8.5**) ● Delay counter (→ refer to **6.8.6**)	● One-shot pulse output(→ refer to **6.9.1**) ● PWM output(→ refer to **6.9.2**) ● Multiple PWM output(→ refer to **6.9.3**) ● Remote control output function (→ refer to **6.9.4**)

Note Only channel 0.

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
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)

6.1 Functions of Timer Array Unit

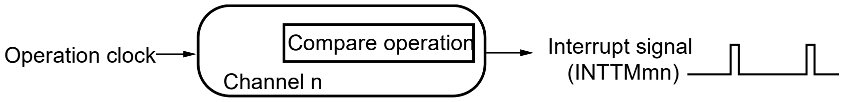
Timer array unit has the following functions.

6.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following purposes without being affected by the operation mode of other channels.

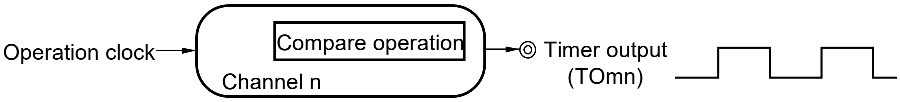
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



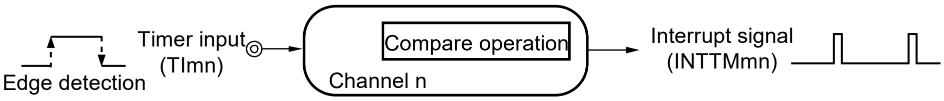
(2) Square wave output

A toggle operation is performed each time INTTMmn interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOmn).



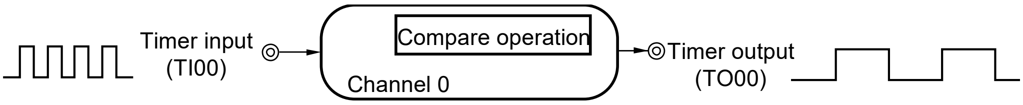
(3) External event counter

Each timer of a unit can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TImn) has reached a specific value.



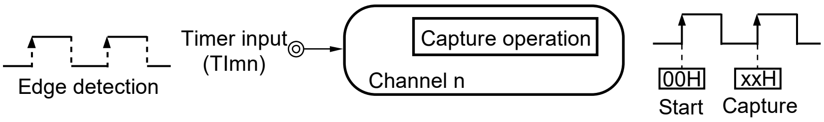
(4) Divider function (channel 0 only)

A clock input from a timer input pin (TI00) is divided and output from an output pin (TO00).



(5) Input pulse interval measurement

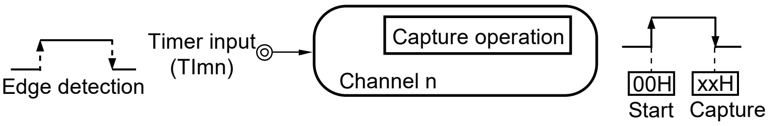
Counting is started by the valid edge of a pulse signal input to a timer input pin (TImn). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



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

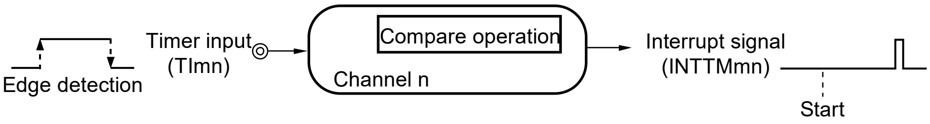
(6) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TImn), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.



(7) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TImn), and an interrupt is generated after any delay period.



Remarks 1 m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

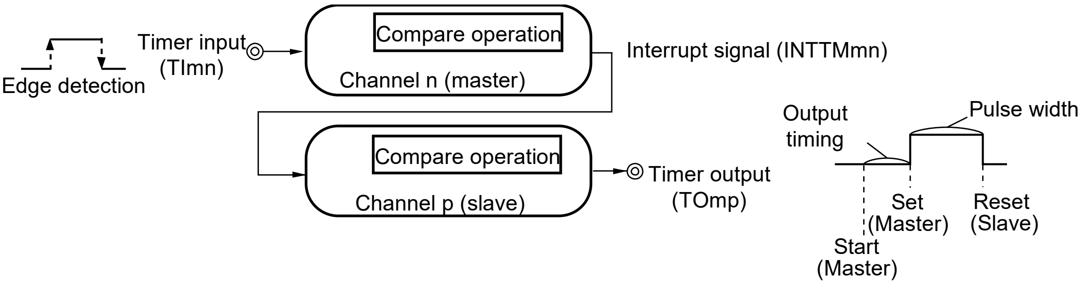
2. The presence or absence of timer I/O pins of channel 0 to 3, 6, 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

6.1.2 Simultaneous channel operation function

By using the combination of a master channel (a reference timer mainly controlling the cycle) and slave channels (timers operating according to the master channel), channels can be used for the following purposes.

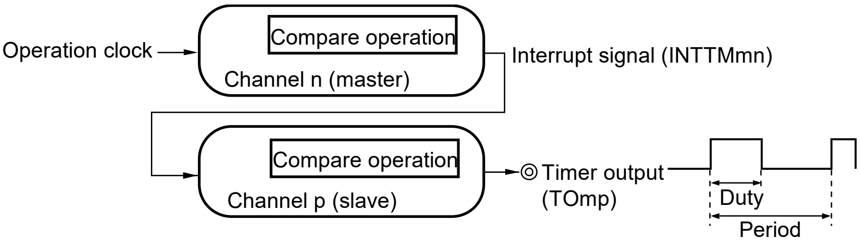
(1) One-shot pulse output

Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.



(2) PWM (Pulse Width Modulation) output

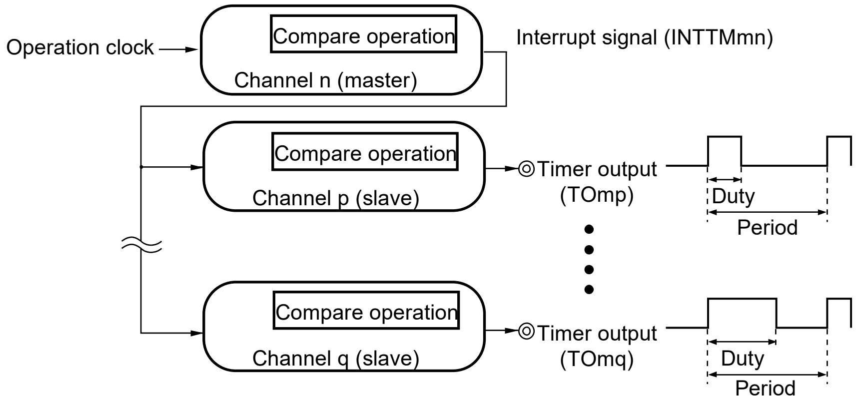
Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



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

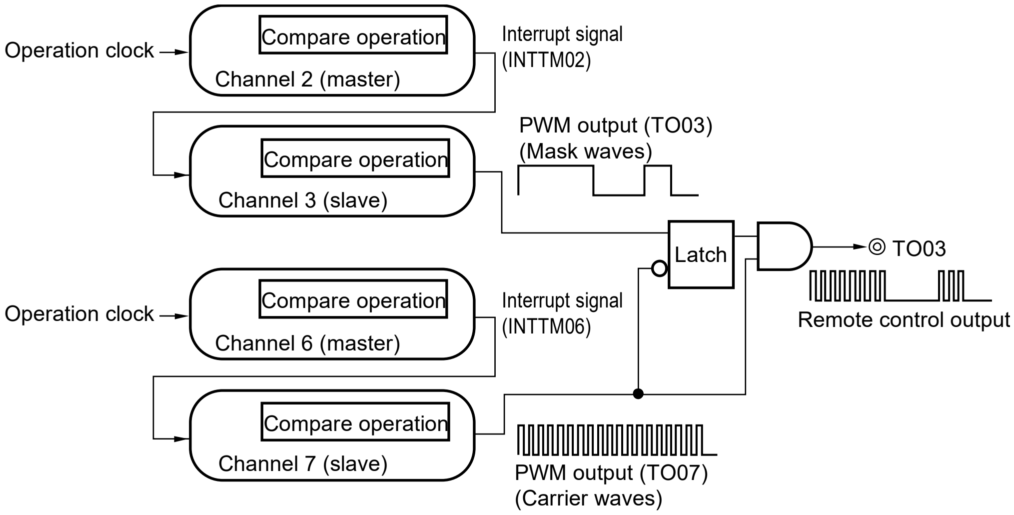
(3) Multiple PWM (Pulse Width Modulation) output

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 6 and 7 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 7 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 3, 6, 7),
p, q: Slave channel number ($n < p < q \leq 7$)

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).

6.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 6-1. Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer count register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI03, TI06, TI07 Note 1
Timer output	TO00 to TO03, TO06, TO07 Note 1 , output controller
Control registers	<Registers of unit setting block> ● Peripheral enable register 0 (PER0) ● Timer clock select register m (TPSm) ● Timer channel enable status register m (TEm) ● Timer channel start register m (TSm) ● Timer channel stop register m (TTm) ● Timer input select register 0 (TIS0) ● Timer output 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) ● Noise filter enable register 1 (NFEN1) ● Port mode register (PMxx) Note 2 ● Port register (Pxx) Note 2

Notes 1. The presence or absence of timer I/O pins of channel 0 to 3, 6, 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

2. The Port mode registers (PMxx) and port registers (Pxx) to be set differ depending on the product. For details, **see 6.3.15 Port mode registers 1, 3, 5, 14 (PM1, PM3, PM5, PM14).**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

The presence or absence of timer I/O pins in each timer array unit channel depends on the product.

Table 6-2. Timer I/O Pins Provided in Each Product

Timer array unit channels	I/O Pins of Each Product	
	48-pin	64-pin
Channel 0	P141/TI00, P140/TO00	
Channel 1	P30/TI01/TO01	
Channel 2	P17/TI02/TO02 (P12)	P17/TI02/TO02 (P54)
Channel 3	P32/TI03/TO03	
Channel 6	–	P51/TI06/TO06
Channel 7	P10/TI07/TO07	P53/TI07/TO07

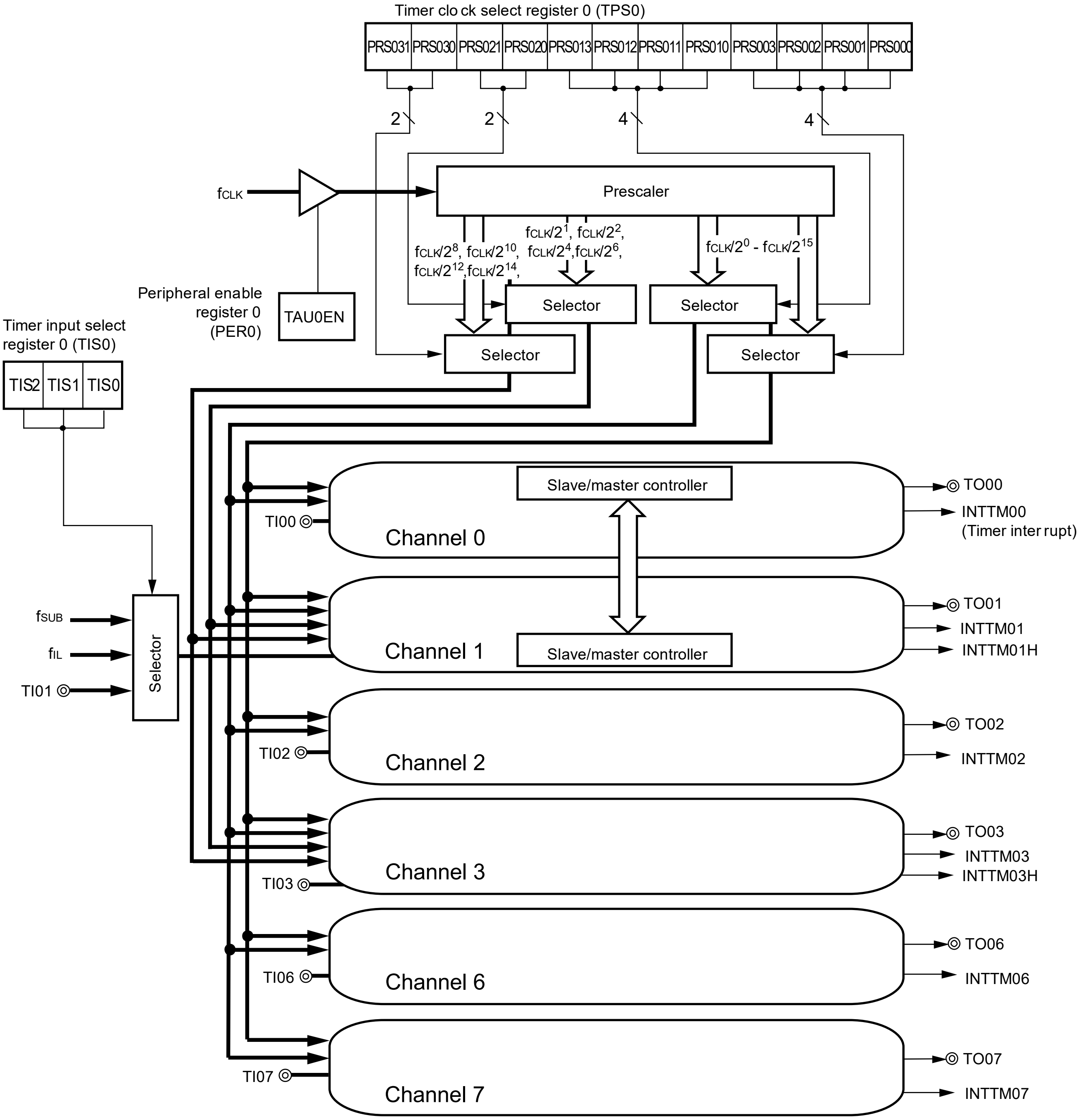
Remarks 1. When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

2. –: There is no timer I/O pin, but the channel is available. (However, the channel can only be used as an interval timer.)

3. "(P12), (P54)" indicates an alternate port when the bit 0 of the peripheral I/O redirection register (PIOR) is set to "1".

Figure 6-1 shows the block diagrams of the timer array unit.

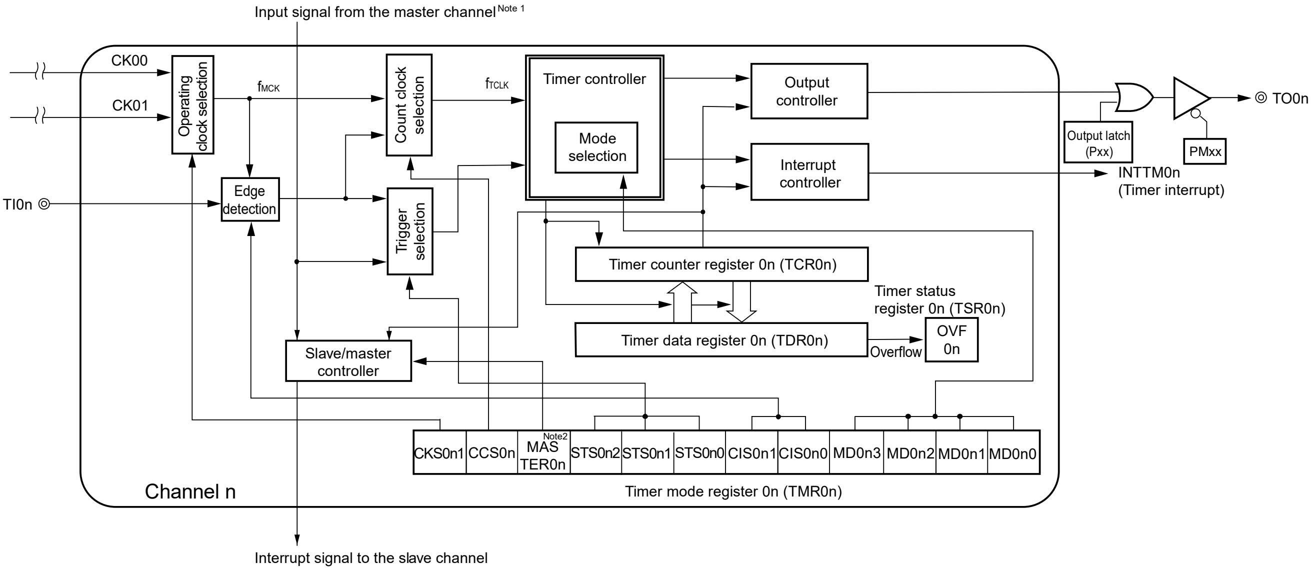
Figure 6-1. Entire Configuration of Timer Array Unit (Example: 64-pin products)



Remark f_{SUB}: Subsystem clock frequency

f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 6-2. Internal Block Diagram of Channels 0, 2, 6 of Timer Array Unit



Notes 1. Channels 2, 6 only

2. n = 2, 6 only

Remark n = 0, 2, 6

Figure 6-3. Internal Block Diagram of Channel 1 of Timer Array Unit

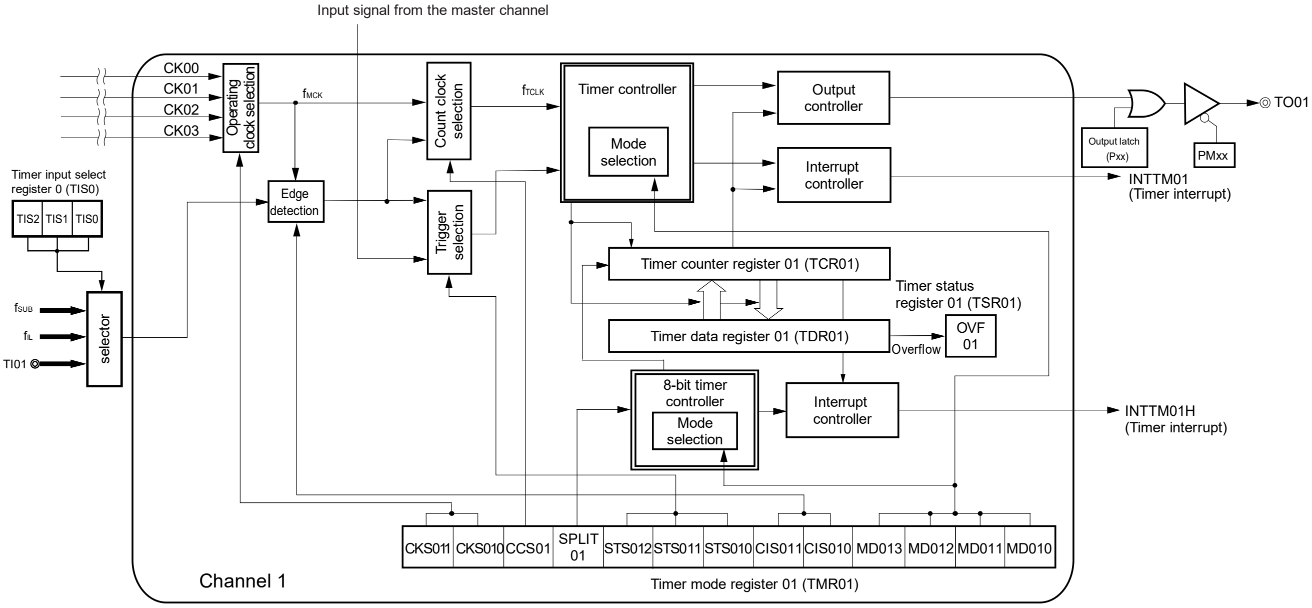


Figure 6-4. Internal Block Diagram of Channel 3 of Timer Array Unit

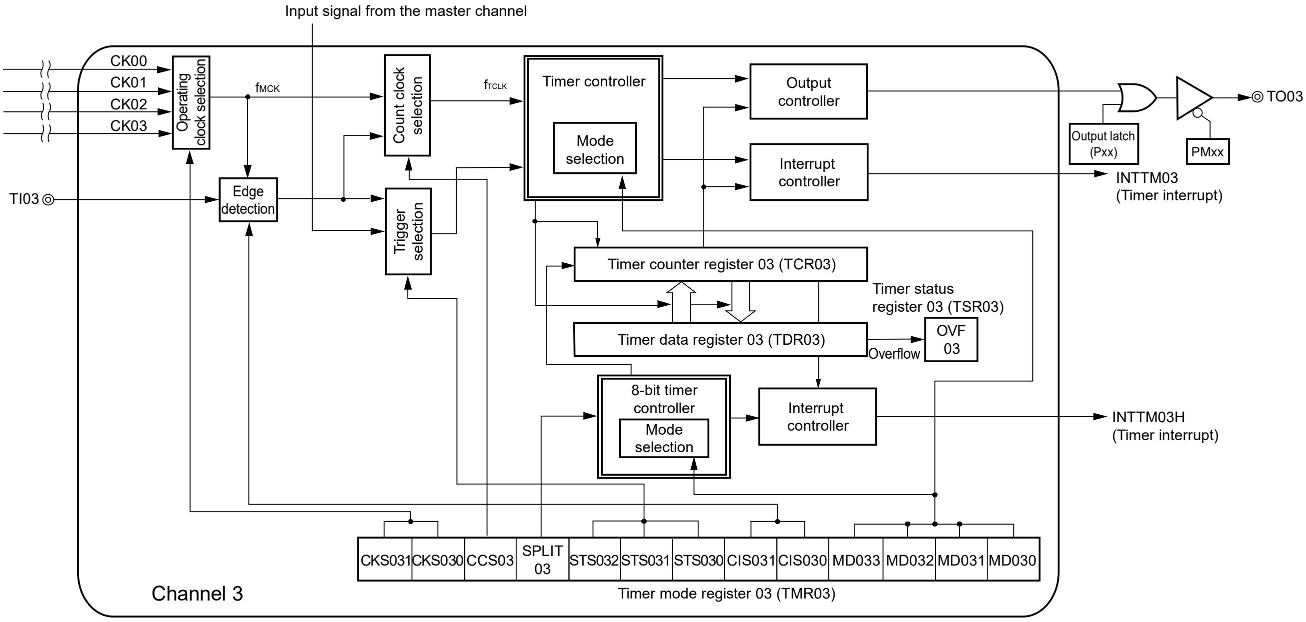
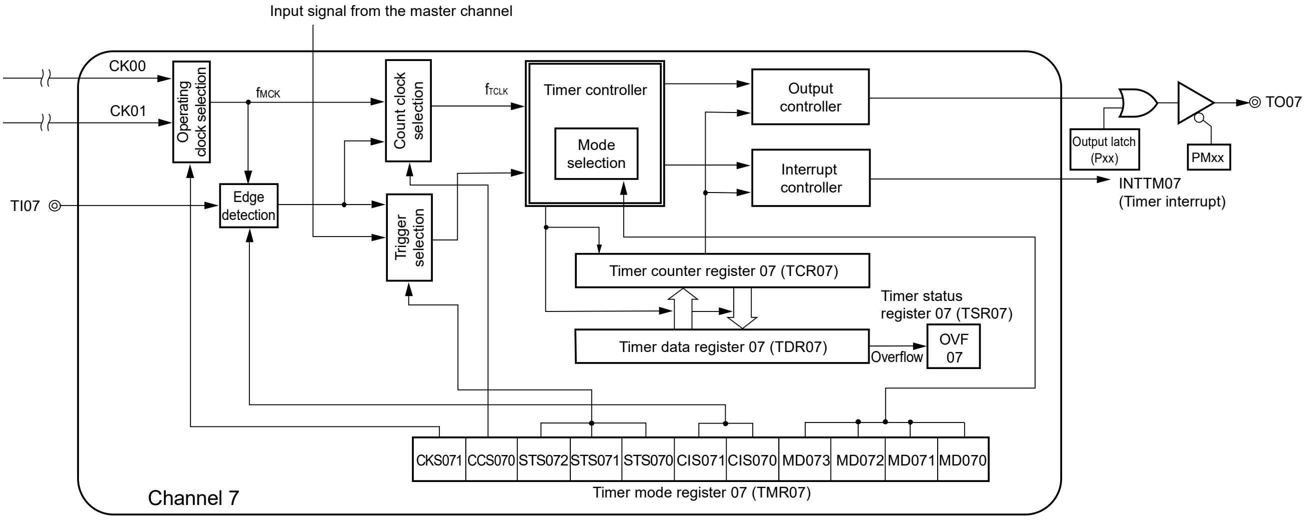


Figure 6-5. Internal Block Diagram of Channel 7 of Timer Array Unit



6.2.1 Timer count register mn (TCRmn)

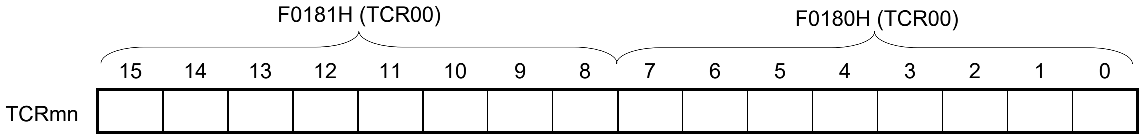
The TCRmn register is a 16-bit read-only register and is used to count clocks.

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 6-6. Format of Timer Count Register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F0186H, F0187H (TCR03), After reset: FFFFH R
F018CH, F018DH (TCR06), F018EH, F018FH (TCR07)



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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 TAU0EN bit of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the slave channel has been completed in the delay count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

The count value is cleared to 0000H in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Caution The count value is not captured to timer data register mn (TDRmn) even when the TCRmn register is read.

The TCRmn register read value differs as follows according to operation mode changes and the operating status.

Table 6-3. Timer Count Register mn (TCRmn) Read Value in Various Operation Modes

Operation Mode	Count Mode	Timer count register mn (TCRmn) Read Value[Note]			
		Value if the operation mode was changed after releasing reset	Value if the Operation was restarted after count operation paused (TTmn = 1)	Value if the operation mode was changed after count operation paused (TTmn = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Count down	FFFFH	Value if stop	Undefined	–
Capture mode	Count up	0000H	Value if stop	Undefined	–
Event counter mode	Count down	FFFFH	Value if stop	Undefined	–
One-count mode	Count down	FFFFH	Value if stop	Undefined	FFFFH
Capture & one-count mode	Count up	0000H	Value if stop	Undefined	Capture value of TDRmn register + 1

Note This indicates the value read from the TCRmn register when channel n has stopped operating as a timer (TEmn = 0) and has been enabled to operate as a counter (TSmn = 1). The read value is held in the TCRmn register until the count operation starts.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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 m1 and m3 (TMRm1, TMRm3) are 1), it is possible to read and write the data in 8-bit units, with TDRm1H and TDRm3H used as the higher 8 bits, and TDRm1L and TDRm3L used as the lower 8 bits.

Reset signal generation clears this register to 0000H.

Figure 6-7. Format of Timer Data Register mn (TDRmn) (n = 0, 2, 6, 7)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02), After reset: 0000H R/W
FFF6CH, FFF6DH (TDR06), FFF6EH, FFF6FH (TDR07)

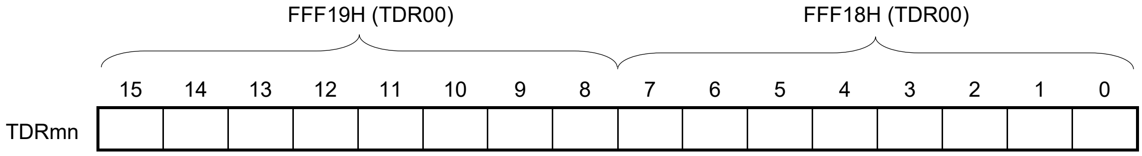
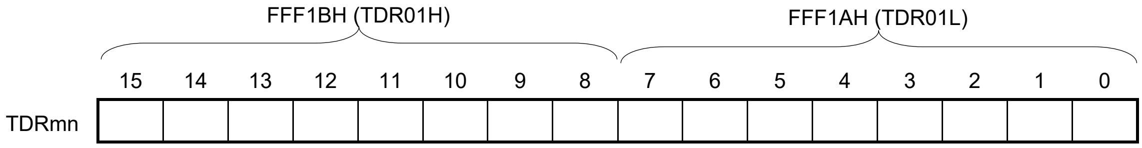


Figure 6-8. Format of Timer Data Register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03) After reset: 0000H 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), n: Channel number (n = 0 to 3, 6, 7)

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register 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)
- Noise filter enable registers 1 (NFEN1)
- 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 6.3.15 Port mode registers 1, 3, 5, 14 (PM1, PM3, PM5, PM14).**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

6.3.1 Peripheral enable register 0 (PER0)

This registers is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the timer array unit is used, be sure to set bit 0 (TAU0EN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 6-9. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

TAU0EN	Control of timer array unit input clock
0	Stops supply of input clock. ● SFR used by the timer array unit cannot be written. ● The timer array unit is in the reset status.
1	Supplies input clock. ● SFR used by the timer array unit can be read/written.

Cautions 1. When setting the timer array unit, be sure to set the following registers first while the TAU0EN bit is set to 1. If TAU0EN = 0, the values of the registers which control the timer array unit are cleared to their initial values and writing to them is ignored (except for the timer input select register 0 (TIS0), timer output select register (TOS), noise filter enable register 1 (NFEN1), port mode registers 1, 3, 5, 14 (PM1, PM3, PM5, PM14), and port registers 1, 3 to 5, 14 (P1, P3 to P5, P14)).

- **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)**

2. Be sure to clear bits 1, 3, 4 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) that are commonly supplied to each channel from external prescaler. CKm1 is selected by using bits 7 to 4 of the TPSm register, and CKm0 is selected by using bits 3 to 0. In addition, for channel 1 and 3, CKm2 is selected by using bits 9 and 8 of the TPSm register, and CKm3 is selected by using bits 13 and 12.

Rewriting of the TPSm register during timer operation is possible only in the following cases.

If the PRSm00 to PRSm03 bits can be rewritten (n = 0 to 3, 6, 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 3, 6, 7):
All channels for which CKm1 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 CKm2 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 0) are stopped (TEmn = 0).
If the PRSm30 and PRSm31 bits can be rewritten (n = 1, 3):
All channels for which CKm3 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 1) are stopped (TEmn = 0).

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

Reset signal generation clears this register to 0000H.

Figure 6-10. Format of Timer Clock Select register m (TPSm) (1/2)

Address: F01B6H, F01B7H 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	PRS m31	PRS m30	0	0	PRS m21	PRS m20	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

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	312.5 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.62 kHz	39.1 kHz	78.1 kHz	156.2 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	9.76 kHz	19.5 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	976 Hz	2.44 kHz	4.88 kHz	9.76 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), stop timer array unit (TTm = 00FFH).

Cautions 1. Be sure to clear bits 15, 14, 11, 10 to “0”.

2. If f_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to 0000H (n = 0 or 1, m = 0 to 3, 6, 7), interrupt requests output from timer array units are not detected.

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. Waveform of the clock to be selected in the TPSm register which becomes high level for one period of f_{CLK} from its rising edge (m = 1 to 15). For details, see **6.5.1 Count clock (f_{TCLK}).**

Figure 6-10. Format of Timer Clock Select register 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	PRS m31	PRS m30	0	0	PRS m21	PRS m20	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

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	312.5 kHz	625 MHz	1.25 MHz	1.5 MHz
1	1	$f_{CLK}/2^{6}$	31.25 kHZ	78.1 kHz	156.2 kHz	312.5 kHz	375 MHz

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	9.76 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 timer array unit (TTm = 00FFH).

The timer array unit must also be stopped if the operating clock (f_{MCK}) specified by using the CKSmn0, and CKSmn1 bits or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Caution Be sure to clear bits 15, 14, 11, 10 to "0".

By using channels 1 and 3 in the 8-bit timer mode and specifying CKm2 or CKm3 as the operation clock, the interval times shown in Table 6-4 can be achieved by using the interval timer function.

Table 6-4. Interval Times Available for Operation Clock CKSm2 or CKSm3

Clock		Interval time (f_{CLK} = 24 MHz)			
		10 μs Note	100 μs Note	1 ms Note	10 ms Note
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%.

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. For details of a signal of $f_{CLK}/2^{j}$ selected with the TPSm register, see **6.5.1 Count clock (f_{TCLK}).**

6.3.3 Timer mode register mn (TMRmn)

The TMRmn register sets an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock, select the master/slave, select the 16 or 8-bit timer (only for channels 1 and 3), specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture and one-count).

Rewriting the TMRmn register is prohibited when the register is in operation (when TEmn = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEmn = 1) (for details, see **6.8 Independent Channel Operation Function of Timer Array Unit** and **6.9 Simultaneous Channel Operation Function of Timer Array Unit**).

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

Reset signal generation clears this register to 0000H.

Caution The bits mounted depend on the channels in the bit 11 of TMRmn register.
TMRm2, TMRm6: MASTERmn bit (n = 2, 6)
TMRm1, TMRm3: SPLITmn bit (n = 1, 3)
TMRm0, TMRm7: Fixed to 0

Figure 6-11. Format of Timer Mode Register mn (TMRmn) (1/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), After reset: 0000H R/W
F019CH, F019DH (TMR06), F019EH, F019FH (TMR07)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 6)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 7)	CKS mn1	CKS mn0	0	CCS mn	0[Note]	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CKS mn1	CKS mn0	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.		

CCS mn	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 When using unit 0. 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.

Cautions 1. Be sure to clear bits 13, 5, and 4 to “0”.

2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for f_{CLK} is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKSmn0 and CKSmn1 bits (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-11. Format of Timer Mode Register mn (TMRmn) (2/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), After reset: 0000H R/W
F019CH, F019DH (TMR06), F019EH, F019FH (TMR07)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 6)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 7)	CKS mn1	CKS mn0	0	CCS mn	0^Note	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

(Bit 11 of TMRmn (n = 2, 6))

MAS TER mn	Selection between using channel n independently or simultaneously with another channel(as a slave or master)
0	Operates in independent channel operation function or as slave channel in simultaneous channel operation function.
1	Operates as master channel in simultaneous channel operation function.
Only the channel 2, 6 can be set as a master channel (MASTERmn = 1). Be sure to use channel 0, 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))

SPLI Tmn	Selection of 8 or 16-bit timer operation for channels 1 and 3
0	Operates as 16-bit timer. (Operates in independent channel operation function or as slave channel in simultaneous channel operation function.)
1	Operates as 8-bit timer.

STS mn2	STS mn1	STS mn0	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), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-11. Format of Timer Mode Register mn (TMRmn) (3/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), After reset: 0000H R/W
F019CH, F019DH (TMR06), F019EH, F019FH (TMR07)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 6)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 7)	CKS mn1	CKS mn0	0	CCS mn	0^{Note}	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CIS mn1	CIS mn0	Selection of TImn pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STSmn2 to STSmn0 bits is other than 010B, set the CISmn1 to CISmn0 bits to 10B.		

MD mn3	MD mn2	MD mn1	MD mn0	Operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	1/0	Interval timer mode	Interval timer/Square wave output/Divider function/PWM output (master)	Counting down
0	1	0	1/0	Capture mode	Input pulse interval measurement	Counting up
0	1	1	0	Event counter mode	External event counter	Counting down
1	0	0	1/0	One-count mode	Delay counter/One-shot pulse output/PWM output (slave)	Counting down
1	1	0	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Counting up
Other than above				Setting prohibited		
The operation of the MDmn0 bit varies depending on each operation mode (see table below).						

Note Bit 11 is fixed at 0 of read only, write is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-11. Format of Timer Mode Register mn (TMRmn) (4/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), After reset: 0000H R/W
F019CH, F019DH (TMR06), F019EH, F019FH (TMR07)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 6)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0, 7)	CKS mn1	CKS mn0	0	CCS mn	0[Note 1]	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above))	MD mn0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0) • Capture mode (0, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode [Note 2] (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated.
	1	Start trigger is valid during counting operation[Note 3]. At that time, interrupt is not generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation.
Other than above		Setting prohibited

Notes 1. Bit 11 is fixed at 0 of read only, write is ignored.

2. In one-count mode, interrupt output (INTTMmn) when starting a count operation and TOmn output are not controlled.

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), n: Channel number (n = 0 to 3, 6, 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 6-12. Format of Timer Status Register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01A6H, F01A7H (TSR03), After reset: 0000H R
F01ACH, F01ADH (TSR06), F01AEH, F01AFH (TSR07)

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

OVF	Counter overflow status of channel n
0	Overflow does not occur.
1	Overflow occurs.
When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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 6-13. 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	0	0	TEm3	TEm2	TEm1	TEm0

TEH03	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 3 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEH01	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEmn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit displays whether operation of the lower 8-bit timer for TEm1 and TEm3 is enabled or stopped when channel 1 or 3 is in the 8-bit timer mode.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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 6-14. 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	0	0	TSm3	TSm2	TSm1	TSm0

TSHm3	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm3 bit is set to 1 and the count operation becomes enabled. The TCRm3 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-6** in **6.5.2 Start timing of counter**).

TSHm1	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm1 bit is set to 1 and the count operation becomes enabled. The TCRm1 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-6** in **6.5.2 Start timing of counter**).

TSmn	Operation enable (start) trigger of channel n
0	No trigger operation
1	The TEmn bit is set to 1 and the count operation becomes enabled. The TCRmn register count operation start in the count operation enabled state varies depending on each operation mode (see **Table 6-6** in **6.5.2 Start timing of counter**). This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for TSm1 and TSm3 when channel 1 or 3 is in the 8-bit timer mode.

Cautions 1. Be sure to clear bits 15 to 12, 10, 8, 5, 4 to “0”

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})

Remarks 1. When the TSm register is read, 0 is always read.

2. m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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, TTHm1, TTHm3 = 0), because they are trigger bits.

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

The lower 8 bits of the TTm register can be set with a 1-bit or 8-bit memory manipulation instruction with TTmL.

Reset signal generation clears this register to 0000H.

Figure 6-15. 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	TTHm 3	0	TTHm 1	0	TTm 7	TTm 6	0	0	TTm 3	TTm 2	TTm 1	TTm 0

TTH m3	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	Operation is stopped (stop trigger is generated).

TTH m1	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	Operation is stopped (stop trigger is generated).

TTm n	Operation stop trigger of channel n
0	No trigger operation
1	TEmn bit clear to 0, to be count operation stop enable status. This bit is the trigger to stop operation of the lower 8-bit timer for TTm1 and TTm3 when channel 1 or 3 is in the 8-bit timer mode.

Caution Be sure to clear bits 15 to 12, 10, 8, 5, 4 of the TTm register to “0”.

Remarks **1.** When the TTm register is read, 0 is always read.

2. m: Unit number (m = 0),n: Channel number (n = 0 to 3, 6, 7)

6.3.8 Timer input select register 0 (TIS0)

The TIS0 register is used to select the channel 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 6-16. Format of Timer Input Select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

Caution High-level width, low-level width of timer input is selected, will require more than $1/f_{MCK}$ +10 ns.

Therefore, when selecting f_{SUB} to f_{CLK} (CSS bit of CKC register = 1), can not set TIS02 bit to 1.

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 wave) to mask the PWM output signal generated by channels 6 and 7 (carrier wave).

Rewriting the TOS register is only possible before counting starts (TE02, TE03, TE06, TE07 = 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 6-17. 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, 6, and 7 is used for timer output)
1	Enable (remote control output to the TO03 pin)

Caution Channels 2, 3, 6, and 7 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 6-18. 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	TOE m7	TOE m6	0	0	TOE m3	TOE m2	TOE m1	TOE m0

TOE mn	Timer output enable/disable of channel n
0	Disable output of timer. Without reflecting on TOmn bit timer operation, to fixed the output. Writing to the TOmn bit is enabled and the level set in the TOmn bit is output from the TOmn pin.
1	Enable output of timer. Reflected in the TOmn bit timer operation, to generate the output waveform. Writing to the TOmn bit is disabled (writing is ignored).

Caution Be sure to clear bits 15 to 8, 5, 4 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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 oh this register can be rewritten by software only when timer output is disabled (TOEmn = 0). When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P140/TO00, P30/TI01/TO01, P17/TI02/TO02, P32/TI03/TO03, P51/TI06/TO06, or P53/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 6-19. 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	TOm 7	TOm 6	0	0	TOm 3	TOm 2	TOm 1	TOm 0

TOm n	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, 5, 4 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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 6-20. 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	TOL m7	TOL m6	0	0	TOL m3	TOL m2	TOL m1	0

TOL mn	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Negative logic output (active-low)

Caution Be sure to clear bits 15 to 8, 5, 4 and 0 to “0”.

Remarks **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.

2. m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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 6-21. 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	TOM m7	TOM m6	0	0	TOM m3	TOM m2	TOM m1	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, 5, 4 and 0 to “0”.

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 3, 6, 7 (n = 0, 2, 6 for master channel)
p: Slave channel number
$n < p \leq 7$
(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 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 input from the TImn pin is selected (CCSmn = 1)** and **6.5.2 Start timing of counter**.

Figure 6-22. Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	0	0	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN07	Enable/disable using noise filter of TI07 pin input signal
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

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

Remark The presence or absence of timer I/O pins of channel 0 to 3, 6, 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

6.3.15 Port mode registers 1, 3, 5, 14 (PM1, PM3, PM5, PM14)

These registers set input/output of ports 1, 3, 5, 14 in 1-bit units.

The presence or absence of timer I/O pins depends on the product. When using the timer array unit, set the following port mode registers according to the product used.

48-pin products: PM1, PM3, PM14
64-pin products: PM1, PM3, PM5, PM14

When using the ports (such as (P140/TO00 and P17/TO02/TI02) to be shared with the timer output pin for timer output, set the port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

Example: When using P17/TO02/TI02 for timer output
Set the PM17 bit of port mode register 1 to 0.
Set the P17 bit of port register 1 to 0.

When using the ports (such as P141/TI00 and P32/TO03/TI03) to be shared with the timer input pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P32/TO03/TI03 for timer input
Set the PM32 bit of port mode register 1 to 1.
Set the P32 bit of port register 1 to 0 or 1.

The PM1, PM3, PM5, PM14 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH

Figure 6-23. Format of Port Mode Registers 1, 3, 5, 14 (PM1, PM3, PM5, PM14) (64-pin products)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	PM32	PM31	PM30

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	1	1	1	PM54	PM53	PM52	PM51	PM50

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	PM147	PM146	PM145	PM144	PM143	PM142	PM141	PM140

PMmn	Pmn pin I/O mode selection (m = 1, 3, 5, 14; n = 0 to 3, 6, 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 1, 3, 5, 14 of the 64-pin products. The format of the port mode register of other products, see **Table 4-3 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product**.

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, 6) can be set as a master channel.
(2) Any channel, except channel 0, can be set as a slave channel.
(3) The slave channel must be lower than the master channel.
  Example: If channel 0 is set as a master channel, channel 1 or those that follow (channels 1, 2, 3, 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 6 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channel 7 cannot be set as the slave channel of master channel 0.

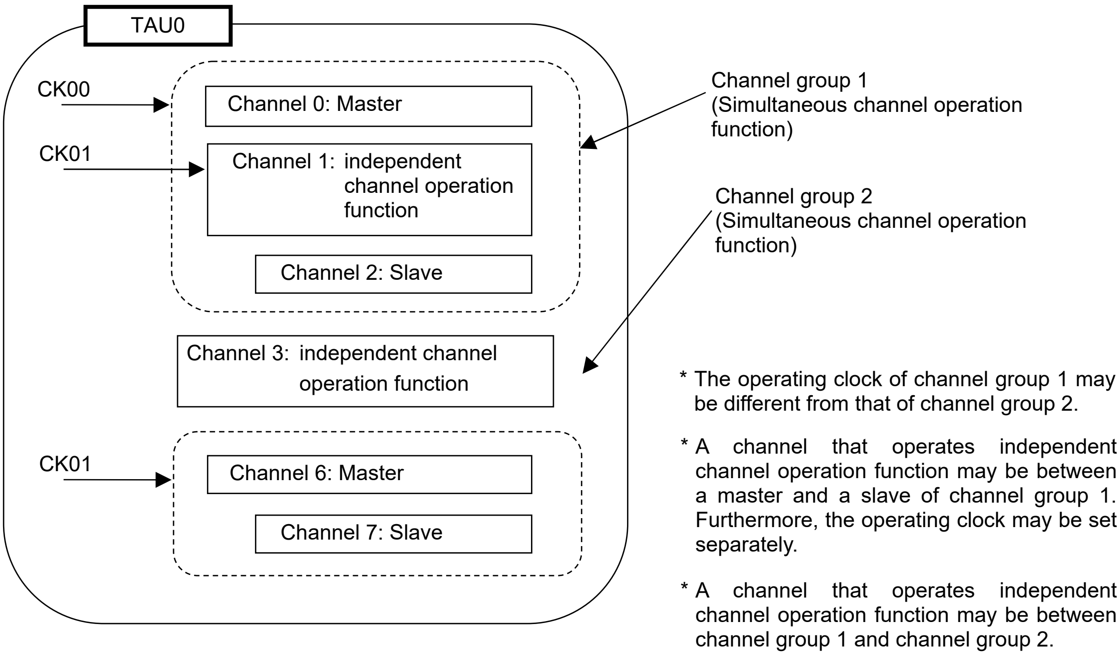
(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKSmn0, CKSmn1 bits (bit 15, 14 of timer mode register mn (TMRmn)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
(7) A master channel can transmit INTTMmn (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use INTTMmn (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or count clock to channels with lower channel numbers.
(9) A master channel cannot use INTTMmn (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.
(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TSmn) of the channels in combination must be set at the same time.
(11) During the counting operation, a TSmn bit of a master channel or TSmn bits of all channels which are operating simultaneously can be set. It cannot be applied to TSmn bits of slave channels alone.
(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TTmn) of the channels in combination must be set at the same time.
(13) CKm2/CKm3 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.
(14) Timer mode register m0 (TMRm0) has no master bit (it is fixed as "0"). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).

If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel operation function in **6.4.1 Basic rules of simultaneous channel operation function** do not apply to the channel groups.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Example



* The operating clock of channel group 1 may be different from that of channel group 2.

* A channel that operates independent channel operation function may be between a master and a slave of channel group 1. Furthermore, the operating clock may be set separately.

* A channel that operates independent channel operation function may be between channel group 1 and channel group 2.

6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

(1) The 8-bit timer operation function applies only to channels 1 and 3.
(2) When using 8-bit timers, set the SPLITmn bit of timer mode register mn (TMRmn) to 1.
(3) The higher 8 bits can be operated as the interval timer function.
(4) At the start of operation, the higher 8 bits output INTTMm1H/INTTMm3H (an interrupt) (which is the same operation performed when MDmn0 is set to 1).
(5) The operation clock of the higher 8 bits is selected according to the CKSmn1 and CKSmn0 bits of the lower-bit TMRmn register.
(6) For the higher 8 bits, the TSHm1/TSHm3 bit is manipulated to start channel operation and the TTHm1/TTHm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEHm1/TEHm3 bit.
(7) The lower 8 bits operate according to the TMRmn register settings. The following three functions support operation of the lower 8 bits:
    - Interval timer function/square waveform 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

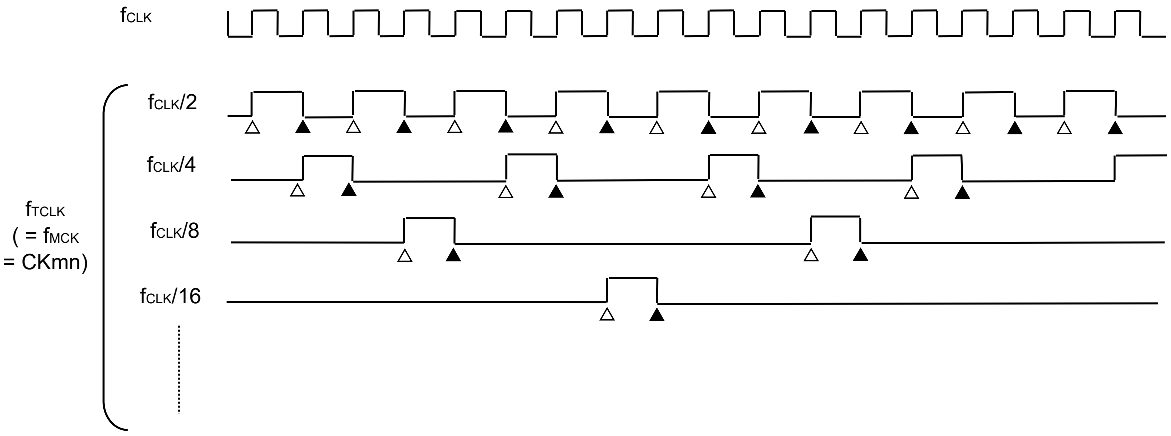
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

(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 TPSm 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 6-24. Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 0)



Remarks 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter

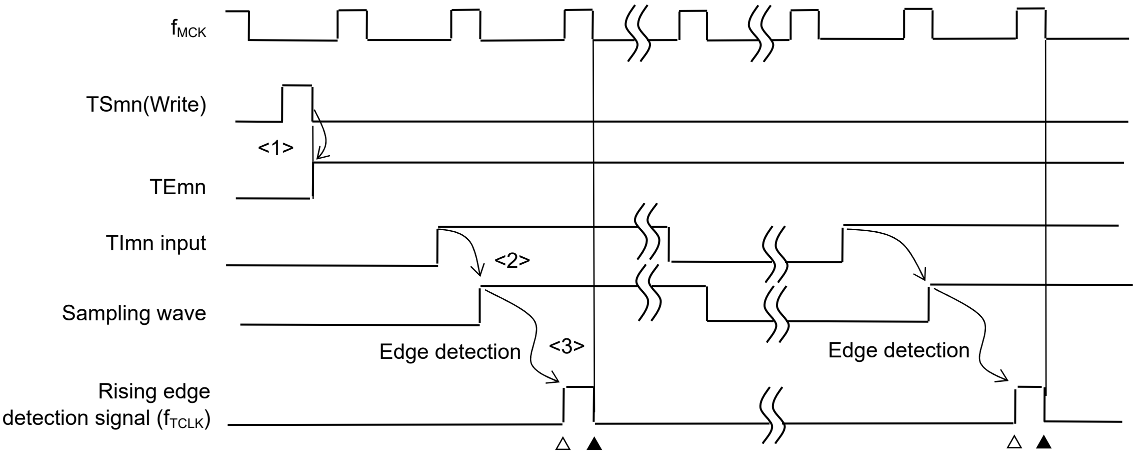
2. f_{CLK}: CPU/peripheral hardware clock

(2) When valid edge of input signal input from the TImn pin is selected (CCSmn = 1)

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 count clock is not a signal which is simply divided f_{CLK} by 2^m, but a signal which becomes high level for one period of f_{CLK} from its rising edge (m = 1 to 15).

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 6-25. 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.

Remarks 1. △ : Rising edge of the count clock

▲ : Synchronization, increment/decrement of counter

2. f_{MCK}: Operation clock of channel n

3. The waveform of the input signal to TImn pin of the input pulse interval measurement, the measurement of high/low width of input signal, the delay counter, the one-shot pulse output is the same as that shown in Figure 6-25.

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 (operation at high-level width measurement)**).

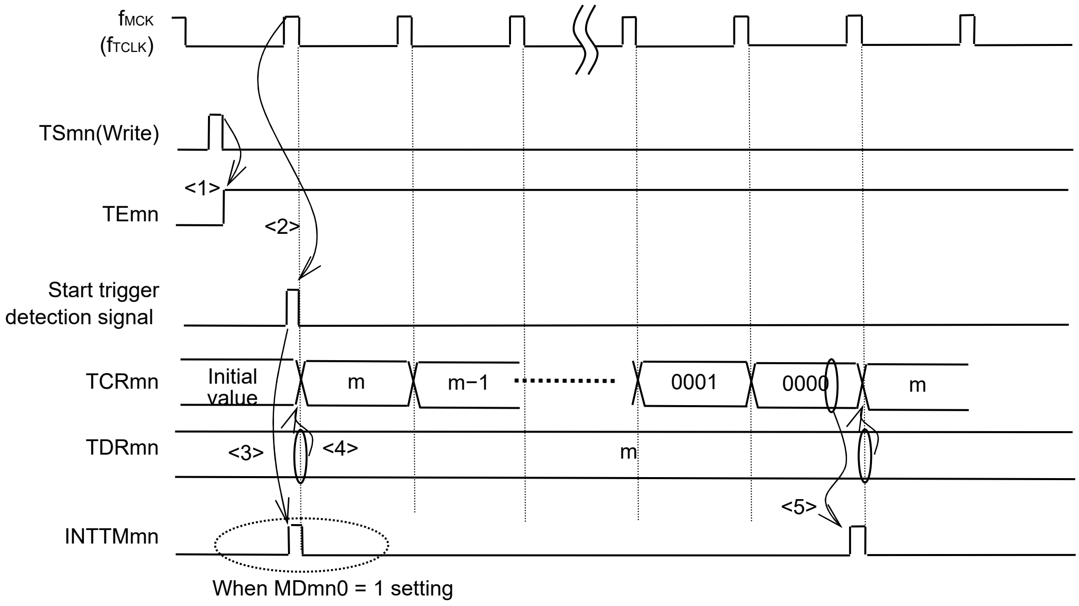
6.5.3 Operation of counter

Here, the counter operation in each mode is explained.

(1) Operation of interval timer mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit. Timer count register mn (TCRmn) holds the initial value until count clock generation.
<2> A start trigger is generated at the first count clock after operation is enabled.
<3> When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.
<4> By the first count clock after the operation enable, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting starts in the interval timer mode.
<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting keeps on.

Figure 6-26. 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

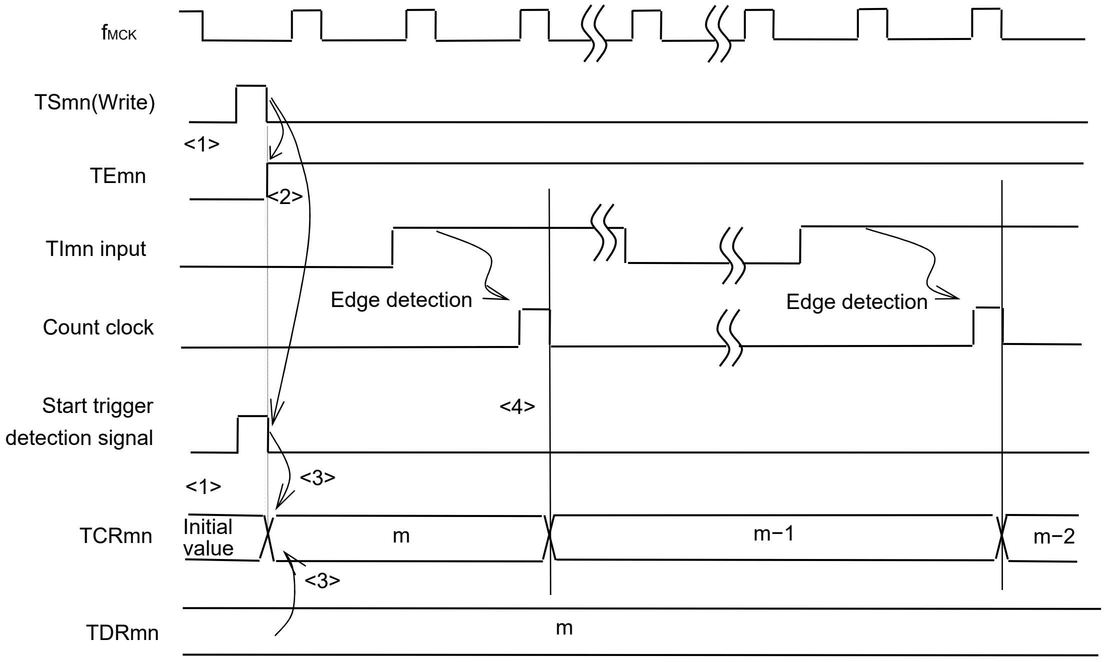
<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).

<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 6-27. Operation Timing (In Event Counter Mode)



Remark The timing is shown in Figure 6-27 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

(3) Operation of capture mode (input pulse interval measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

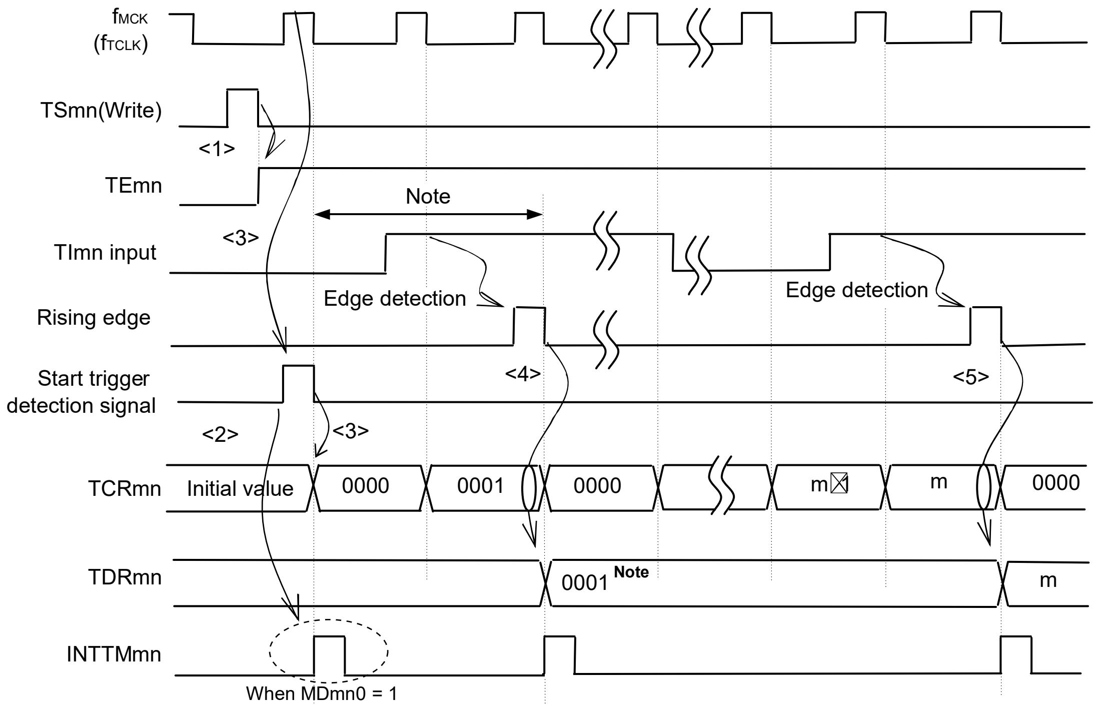
<2> Timer count register mn (TCRmn) holds the initial value until count clock generation.

<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 nomeaning. 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 6-28. 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 timing is shown in Figure 6-28 indicates while 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.

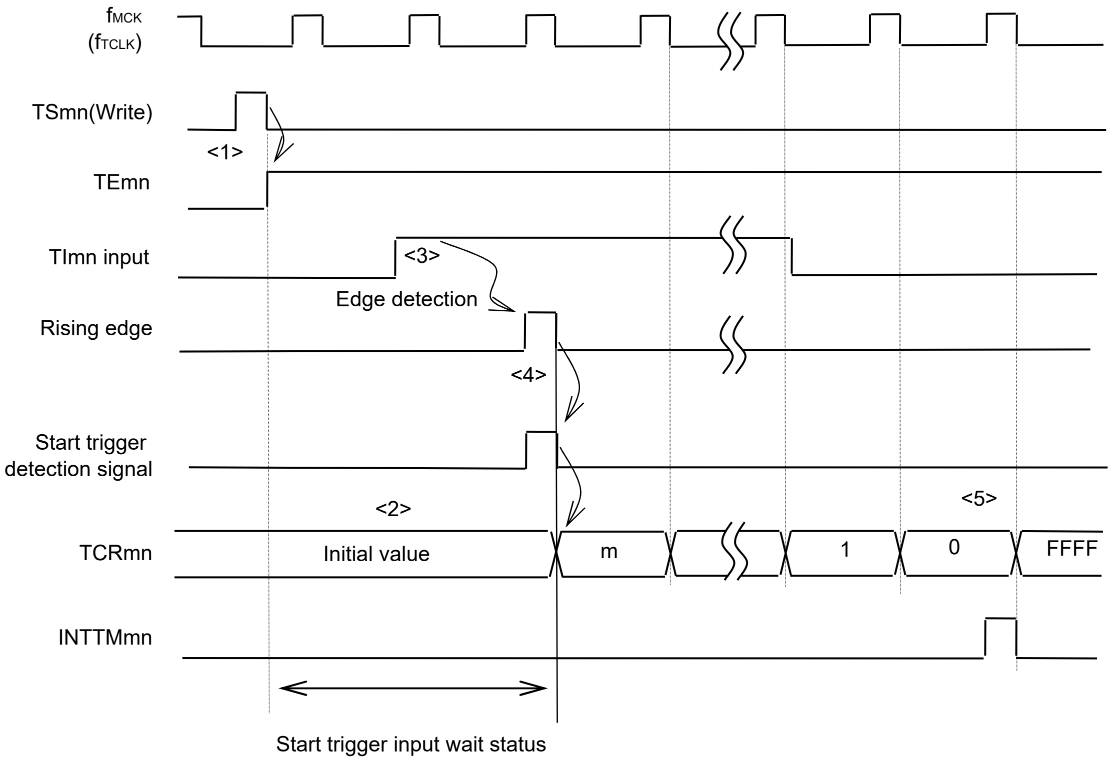
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<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 6-29. Operation Timing (In One-count Mode)

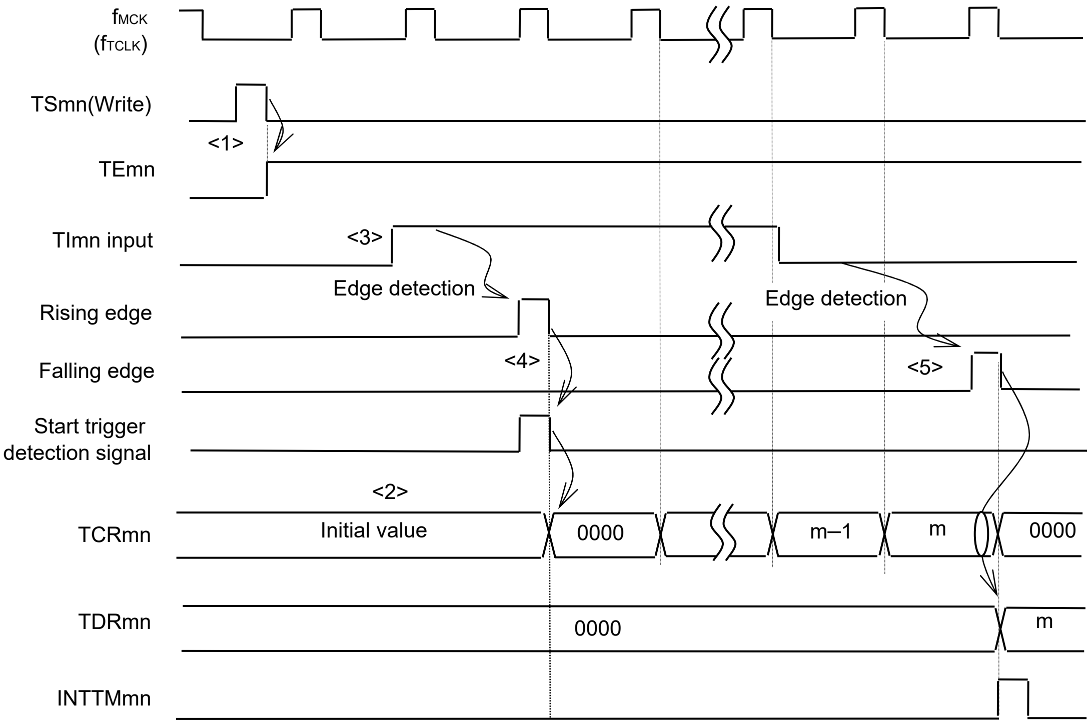


Remark The timing is shown in Figure 6-29 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

(5) Start timing in capture & one-count mode (operation at high-level width measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit of timer channel start register m (TSm).
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.
<3> Rising edge of the TImn input is detected.
<4> On start trigger detection, the value of 0000H is loaded to the TCRmn register and count starts.
<5> On detection of the falling edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 6-30. Start Timing (In Capture & One-count Mode)

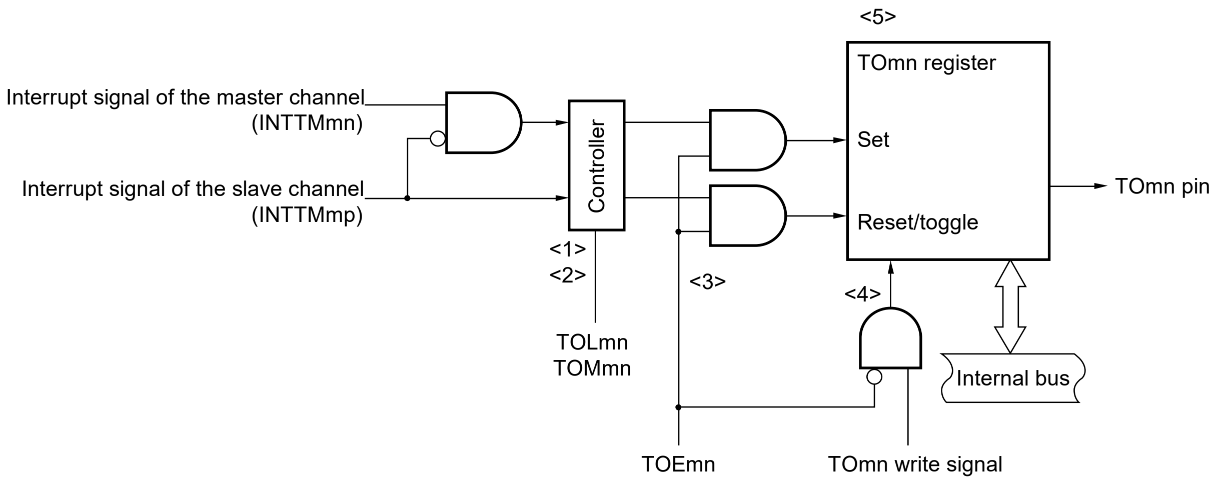


Remark The timing is shown in Figure 6-30 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the 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 6-31. Output Circuit Configuration



The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (master channel output mode), the set value of timer output level register m (TOLm) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register m (TOm).

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:

When TOLmn = 0: Positive logic output (INTTMmn → set, INTTM0p → reset)
When TOLmn = 1: Negative logic output (INTTMmn → reset, INTTM0p → set)

When INTTMmn and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register. Writing to the TOm register (TOmn write signal) becomes invalid.
When TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set timer operation stopped (TOEmn = 0) and to write a value to the TOm register.

<4> While timer output is disabled (TOEmn = 0), writing to the TOmn bit to the target channel (TOmn write signal) becomes valid. When timer output is disabled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TOm register.

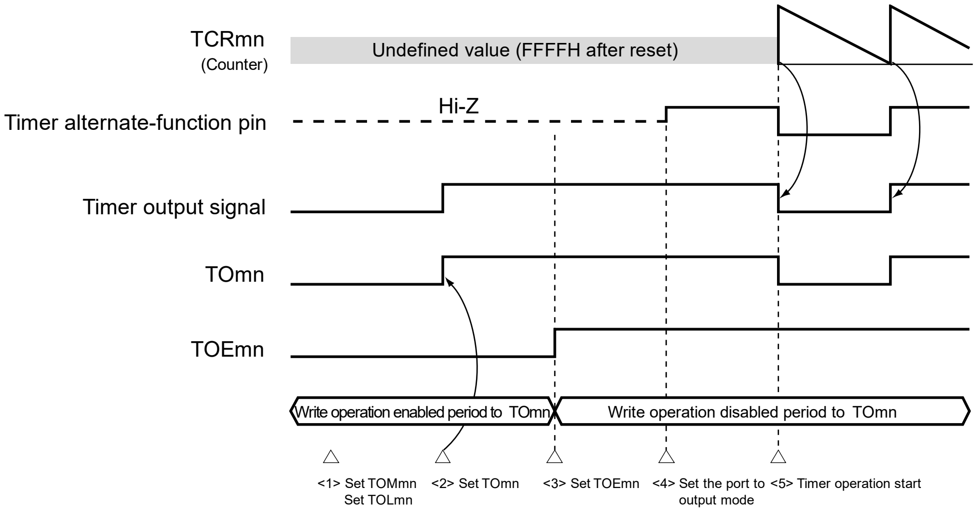
<5> The TOm register can always be read, and the TOmn pin output level can be checked.

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 3, 6, 7 (n = 0, 2, 6 for master channel)
p: Slave channel number
$n < p \leq 7$

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 6-32. 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 I/O setting is set to output (see 6.3.15 Port mode registers 1, 3, 5, 14 (PM1, PM3, PM5, PM14).).
<5> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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, and TOLm registers to the values stated in the register setting example of each operation described in **6.7** and **6.8**.

When the values set to the TOEm, and TOLm registers (but not the TOm register) are changed close to the occurrence of the timer interrupt (INTTMmn) of each channel, the waveform output to the TOmn pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMmn) occurs.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

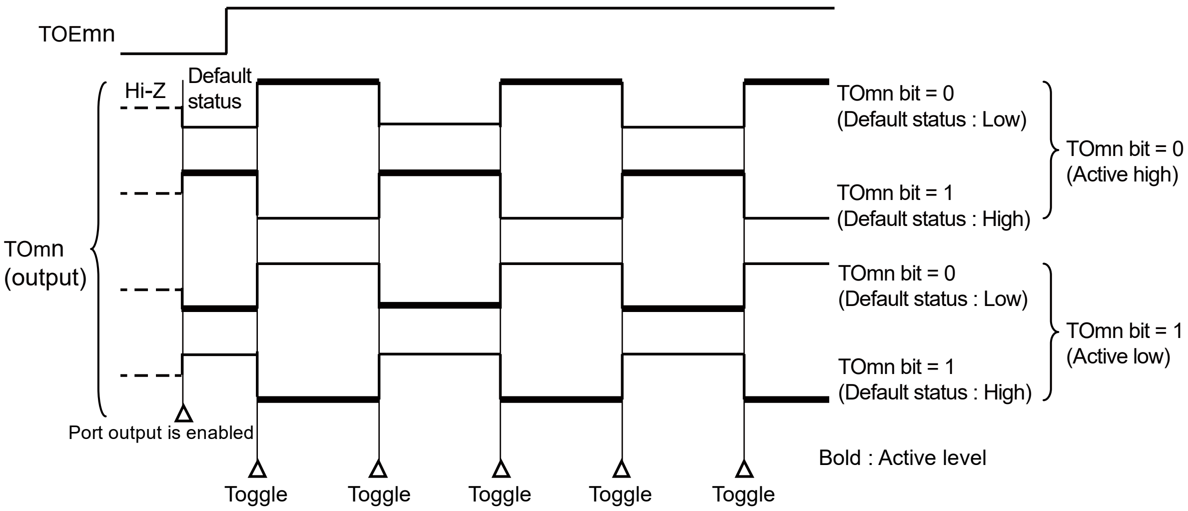
(2) Default level of TOmn pin and output level after timer operation start

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 6-33. TOmn Pin Output Status at Toggle Output (TOMmn = 0)



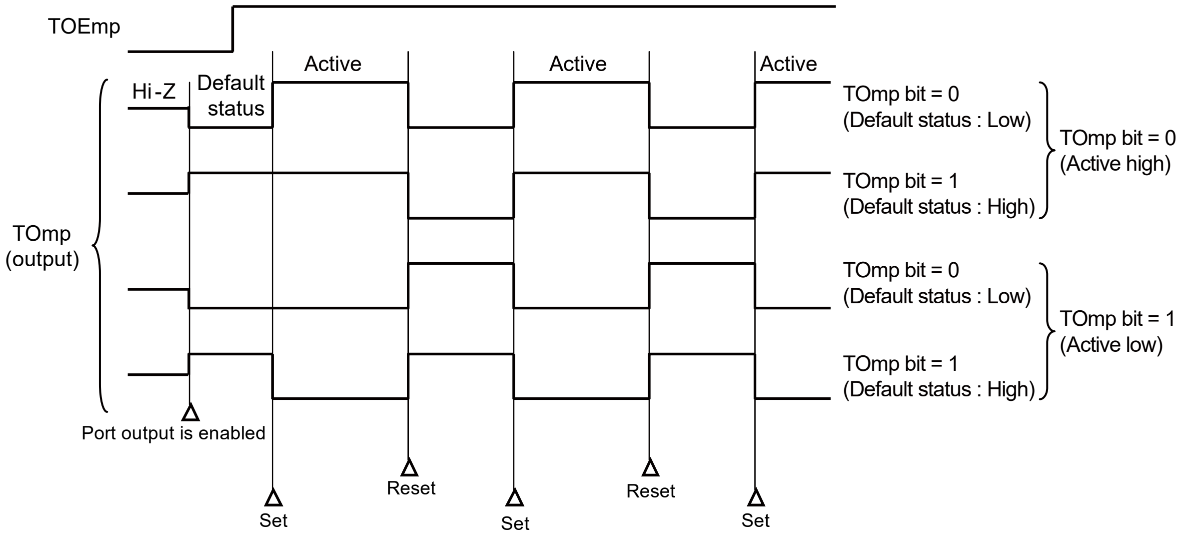
Remarks 1. Toggle: Reverse TOmn pin output status

2. m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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 6-34. TOmn Pin Output Status at PWM Output (TOMmp = 1)



Remarks 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.

2. m: Unit number (m = 0), n: Channel number (p = 1 to 3, 6, 7)

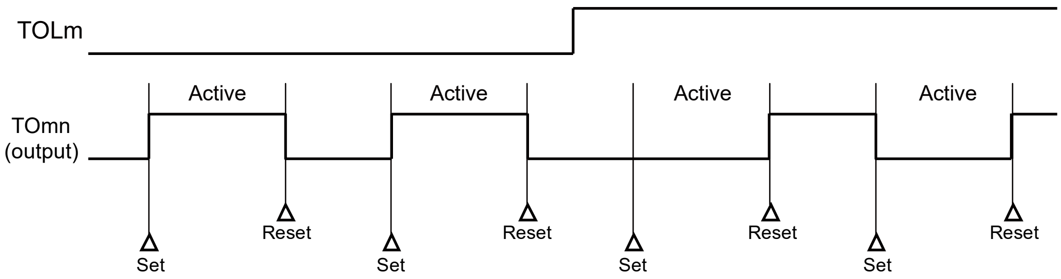
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(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 6-35. Operation when TOLm Register Has Been Changed during Timer Operation



Remarks 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.

2. m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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.

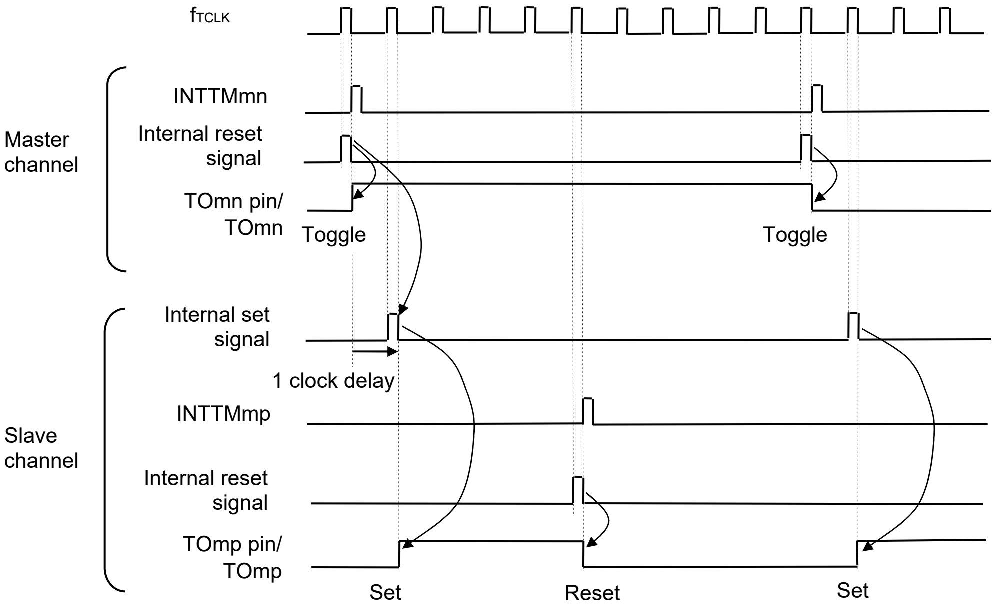
Figure 6-36 shows the set/reset operating statuses where the master/slave channels are set as follows.

Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0

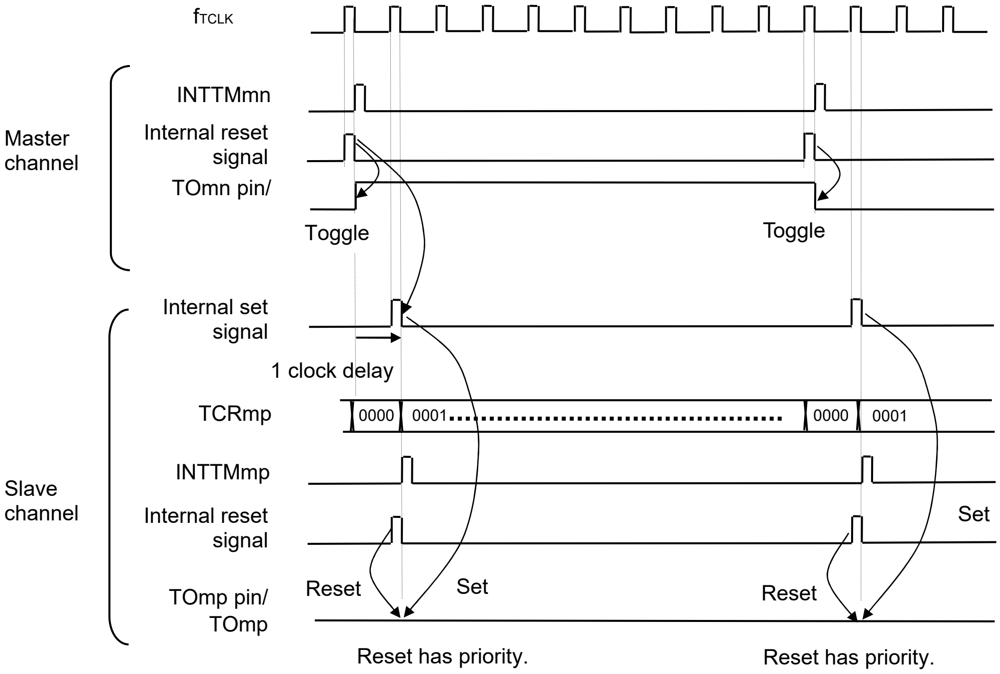
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 6-36. Set/Reset Timing Operating Statuses

(1) Basic operation timing



(2) Operation timing when 0% duty



Remarks 1. Internal reset signal: TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

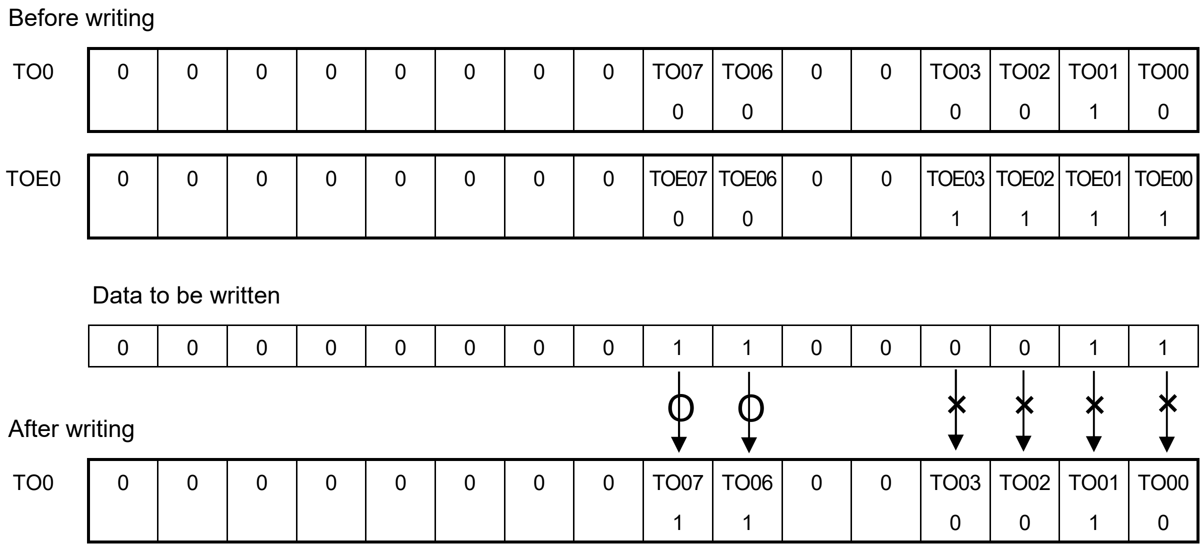
2. m: Unit number (m = 0)
n: Channel number (n = 0 to 3, 6, 7)
n = 0 to 3, 6, 7 (n = 0, 2, 6 for master channel)
p: Slave channel number
$n < p \leq 7$

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 6-37 Example of TO0n Bit Collective Manipulation



Before writing

TO0	0	0	0	0	0	0	0	0	TO07 0	TO06 0	0	0	TO03 0	TO02 0	TO01 1	TO00 0
TOE0	0	0	0	0	0	0	0	0	TOE07 0	TOE06 0	0	0	TOE03 1	TOE02 1	TOE01 1	TOE00 1

Data to be written

0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1

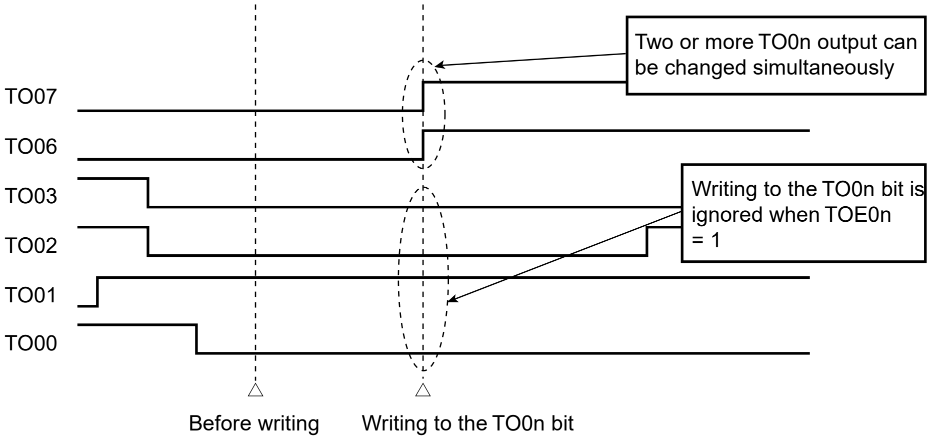
After writing

TO0	0	0	0	0	0	0	0	0	TO07 1	TO06 1	0	0	TO03 0	TO02 0	TO01 1	TO00 0

Writing is done only to the TOmn bit with TOEmn = 0, and writing to the TOmn bit with TOEmn = 1 is ignored.

TOmn (channel output) to which TOEmn = 1 is set is not affected by the write operation. Even if the write operation is done to the TOmn bit, it is ignored and the output change by timer operation is normally done.

Figure 6-38. TO0n Pin Statuses by Collective Manipulation of TO0n Bit



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

6.6.5 Timer Interrupt and TOmn Pin Output at Operation Start

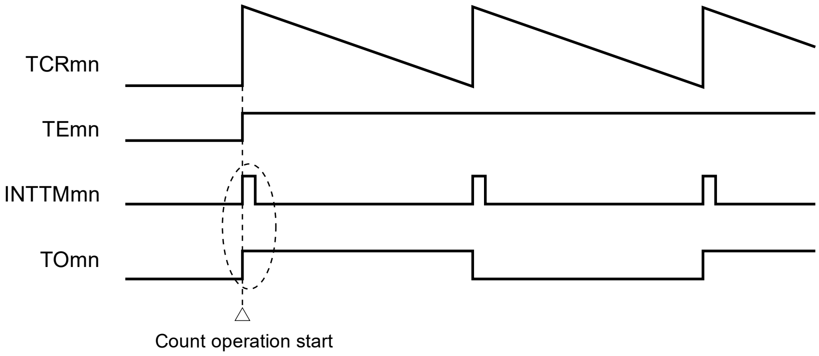
In the interval timer mode or capture mode, the MDmn0 bit in timer mode register mn (TMRmn) sets whether or not to generate a timer interrupt at count start.

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.

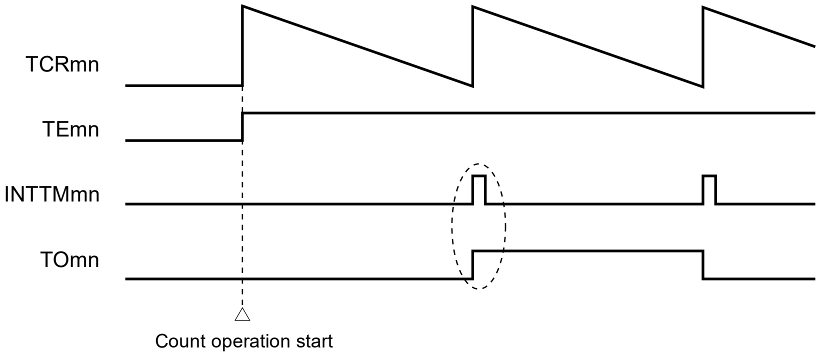
Figure 6-39 show operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 6-39. When MDmn0 is set to 1



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

Figure 6-40. When MDmn0 is set to 0



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOmn does not change either. After counting one cycle, INTTMmn is output and TOmn performs a toggle operation.

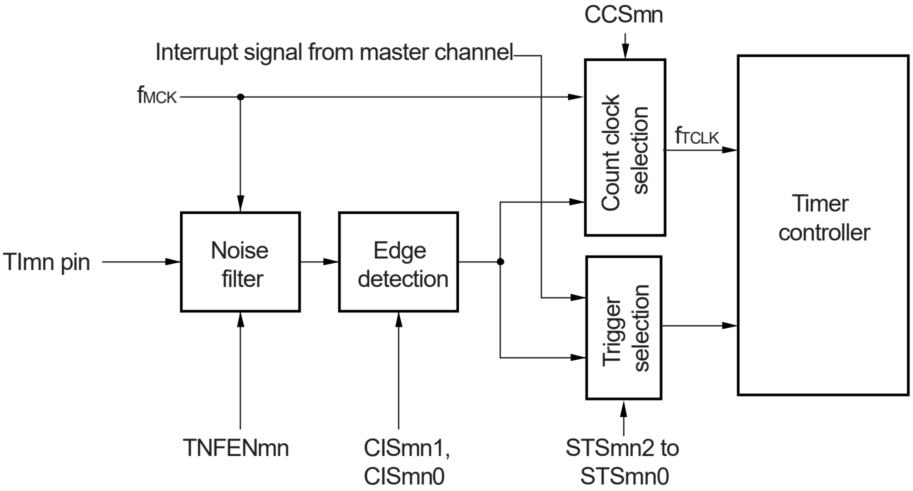
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

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.

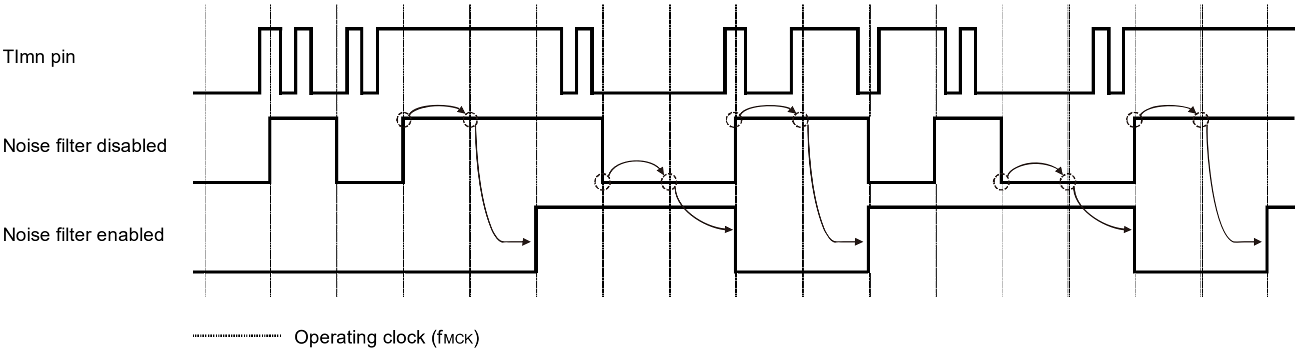
Figure 6-41. Input Circuit Configuration



6.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for channel n. When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for channel n, whether the signal keeps the same value for two clock cycles is detected. The following shows differences in waveforms output from the noise filter between when the noise filter is enabled and disabled.

Figure 6-42. Sampling Waveforms through TImn Input Pin with Noise Filter Enabled and Disabled



6.7.3 Cautions on channel input operation

When a timer input pin is set as unused, the operating clock is not supplied to the noise filter. Therefore, after settings are made to use the timer input pin, the following wait time is necessary before a trigger is specified to enable operation of the channel corresponding to the timer input pin.

(1) Noise filter is disabled

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are 0 and then one of them is set to 1, wait for at least two cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

(2) Noise filter is enabled

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are all 0 and then one of them is set to 1, wait for at least four cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

6.8 Independent Channel Operation Function of Timer Array Unit

6.8.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTMmn (timer interrupt) at fixed intervals.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

(2) Operation as square wave output

TOmn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOmn can be calculated by the following expressions.

- Period of square wave output from TOmn = Period of count clock × (Set value of TDRmn + 1) × 2
- Frequency of square wave output from TOmn = Frequency of count clock/{(Set value of TDRmn + 1) × 2}

Timer count register mn (TCRmn) operates as a down counter in the interval timer mode.

The TCRmn register loads the value of timer data register mn (TDRmn) at the first count clock after the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1. If the MDmn0 bit of timer mode register mn (TMRmn) is 0 at this time, INTTMmn is not output and TOmn is not toggled. If the MDmn0 bit of the TMRmn register is 1, INTTMmn is output and TOmn is toggled.

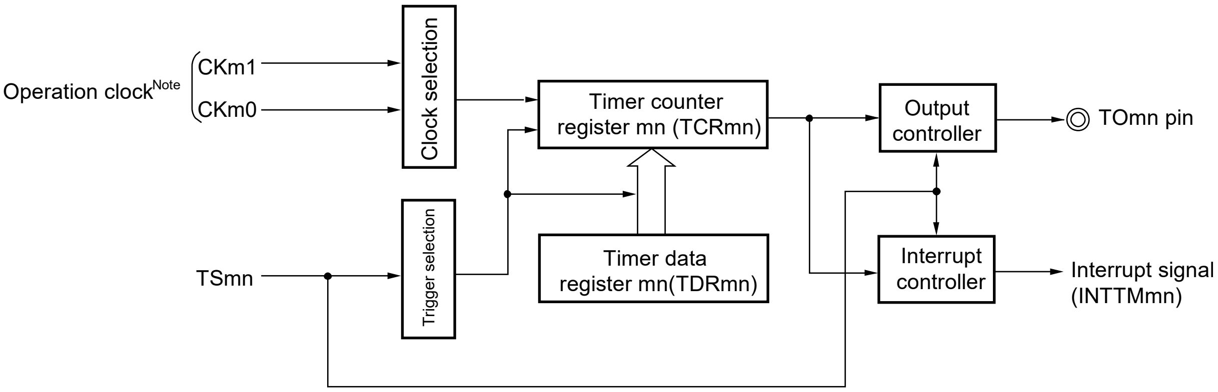
After that, the TCRmn register count down in synchronization with the count clock.

When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, the TCRmn register loads the value of the TDRmn register again. After that, the same operation is repeated.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

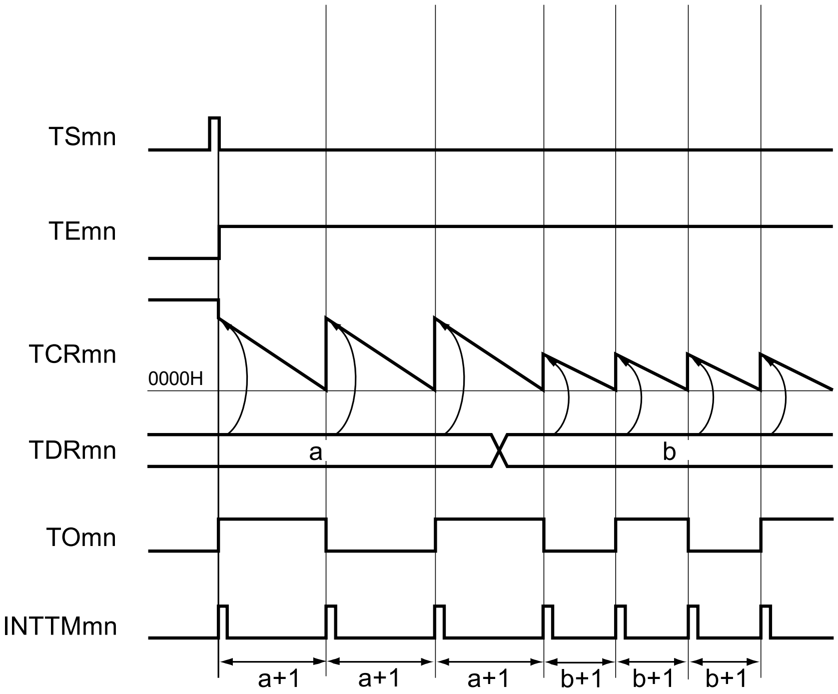
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-43. 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 6-44. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)



Remarks **1.** m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

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 6-45. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0/1	0	0	0	0	0	0	0	0	0	0	1/0

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.
1: Generates INTTMmn and inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channels 2, 6)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
1: 8-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	1/0

0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	1/0

0: Stops the TOmn output operation by counting operation.
1: Enables the TOmn output operation by counting operation.

Note	
TMRm2, TMRm6:	MASTERmn bit
TMRm1, TMRm3:	SPLITmn bit
TMRm0, TMRm7:	Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-45. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/2)

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-46. Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 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. Clears the port register and port mode register to 0.	TOmn does not change because channel stops operating. The TOmn pin outputs the TOmn set level.
Operation start	(Sets the TOEmn bit to 1 only if using TOmn output and resuming operation.). Sets the TSmn (TSHm1, TSHm3) bit to 1. The TSmn (TSHm1, TSHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3) = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn) at the count clock input. INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set values of the TMRmn register, TOMmn, and TOLmn bits cannot be changed. Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOm and TOEm registers can be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn (TTHm1, TTHm3) bit is set to 1. The TTmn (TTHm1, TTHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3), and count operation stops. The TCRmn register holds count value and stops. The TOmn output is not initialized but holds current status.
	The TOEmn bit is cleared to 0 and value is set to the TOmn bit.	The TOmn pin outputs the TOmn bit set level.

Operation is resumed.

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

Figure 6-46. 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. When holding the TOmn pin output level is not necessary Setting not required.	The TOmn pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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) of timer channel start register m (TSm) to 1.

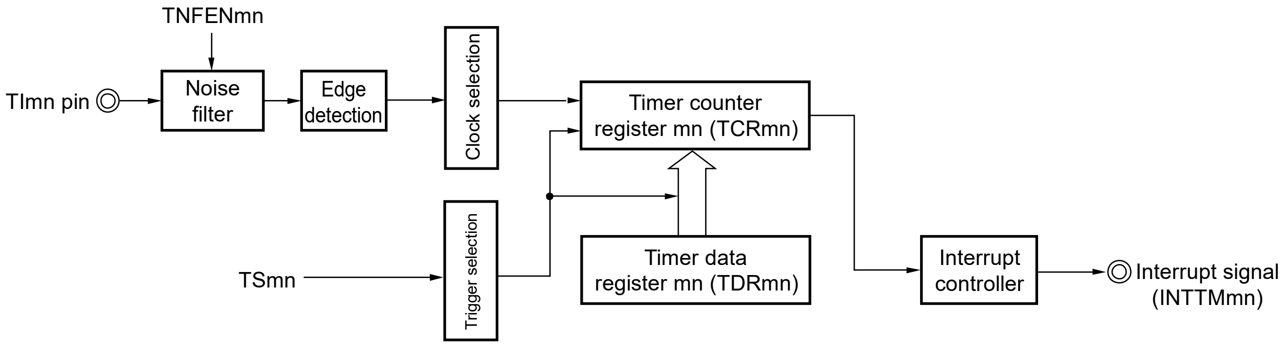
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.

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.

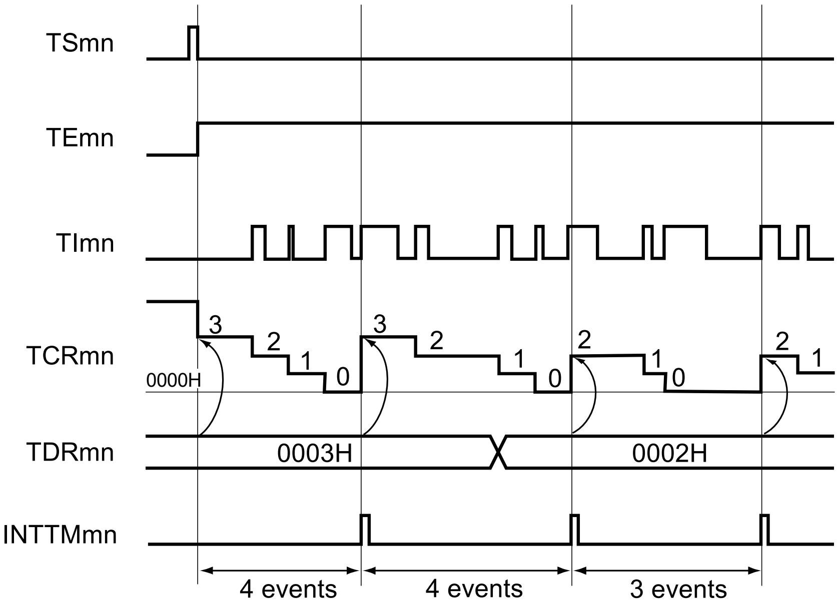
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 6-47. Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-48. Example of Basic Timing of Operation as External Event Counter



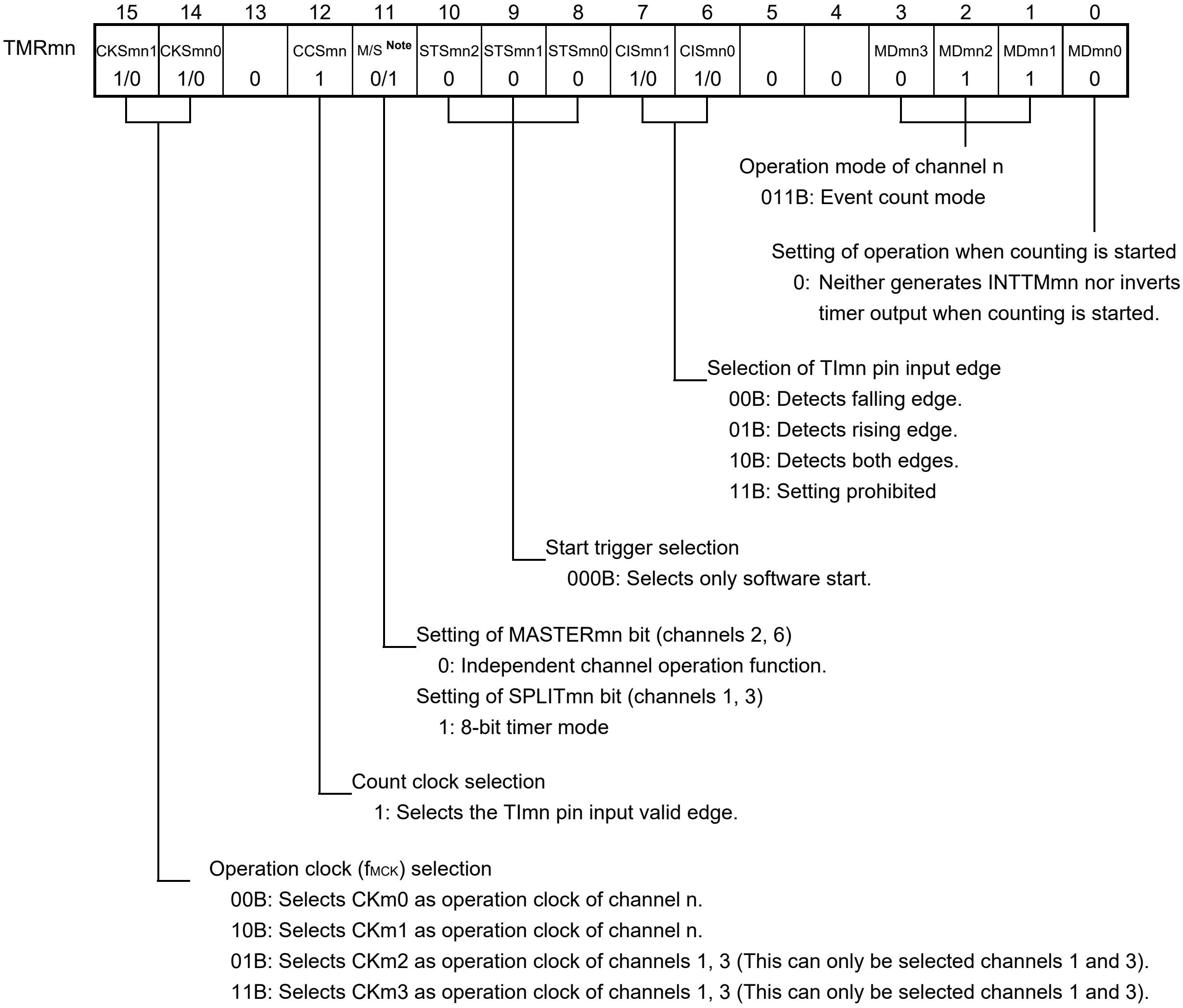
Remarks **1.** m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

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 6-49. Example of Set Contents of Registers in External Event Counter Mode (1/2)

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	1	0/1	0	0	0	1/0	1/0	0	0	0	1	1	0

Operation mode of channel n
011B: Event count mode

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channels 2, 6)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
1: 8-bit timer mode

Count clock selection
1: Selects the TImn pin input valid edge.

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops the TOmn output operation by counting operation.

Note		
	TMRm2, TMRm6:	MASTERmn bit
	TMRm1, TMRm3:	SPLITmn bit
	TMRm0, TMRm7:	Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-49. Example of Set Contents of Registers in External Event Counter Mode (2/2)

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-50. Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	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. Sets corresponding bit of noise filter enable register 1, 2 (NFEN1, NFEN2) to 1. 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 TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

6.8.3 Operation as frequency divider (channel 0 only)

The timer array unit can be used as a frequency divider that divides a clock input to the TI00 pin and outputs the result from the TO00 pin.

The divided clock frequency output from TO00 can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDR00 + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≈ Input clock frequency/(Set value of TDR00 + 1)

Timer count register 00 (TCR00) operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS00) of timer channel start register 0 (TS0) is set to 1, the TCR00 register loads the value of timer data register 00 (TDR00) when the TI00 valid edge is detected.

If the MD000 bit of timer mode register 00 (TMR00) is 0 at this time, INTTM00 is not output and TO00 is not toggled. If the MD000 bit of timer mode register 00 (TMR00) is 1, INTTM00 is output and TO00 is toggled.

After that, the TCR00 register counts down at the valid edge of the TI00 pin. When TCR00 = 0000H, it toggles TO00. At the same time, the TCR00 register loads the value of the TDR00 register again, and continues counting.

If detection of both the edges of the TI00 pin is selected, the duty factor error of the input clock affects the divided clock period of the TO00 output.

The period of the TO00 output clock includes a sampling error of one period of the operation clock.

Clock period of TO00 output = Ideal TO00 output clock period ± Operation clock period (error)

The TDR00 register can be rewritten at any time. The new value of the TDR00 register becomes valid during the next count period.

Figure 6-51. Block Diagram of Operation as Frequency Divider

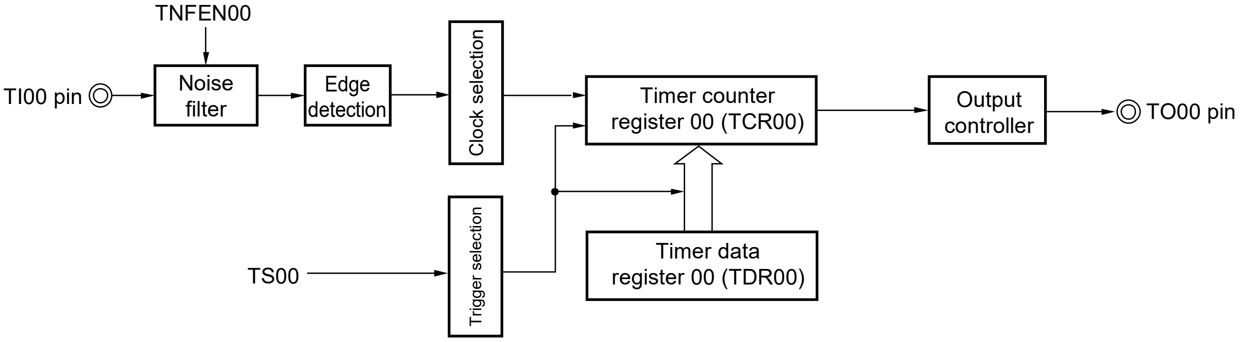
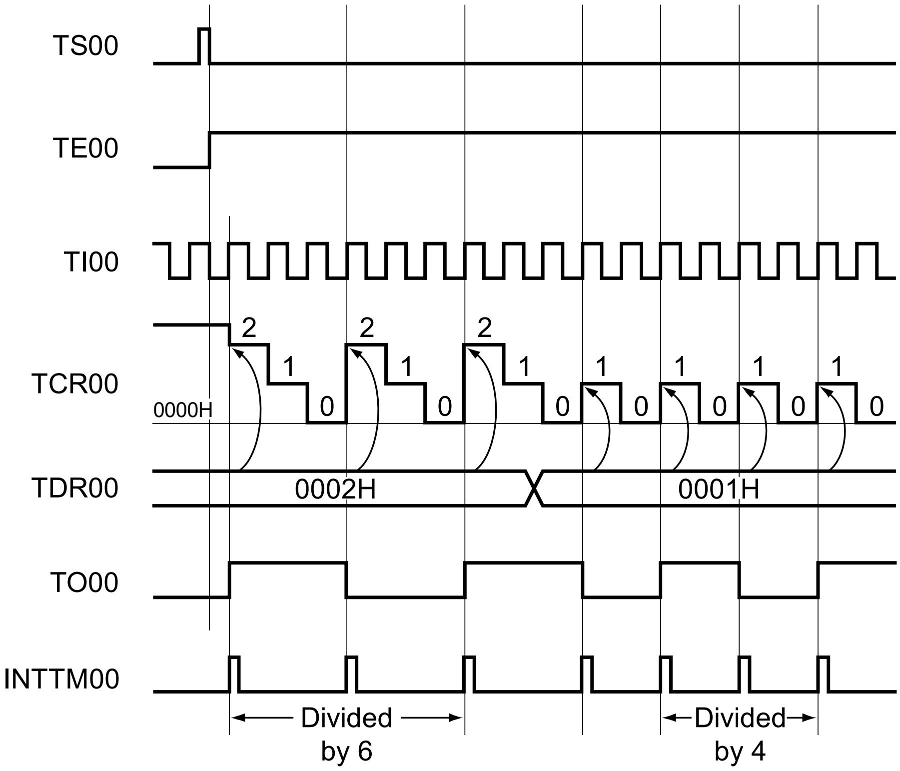


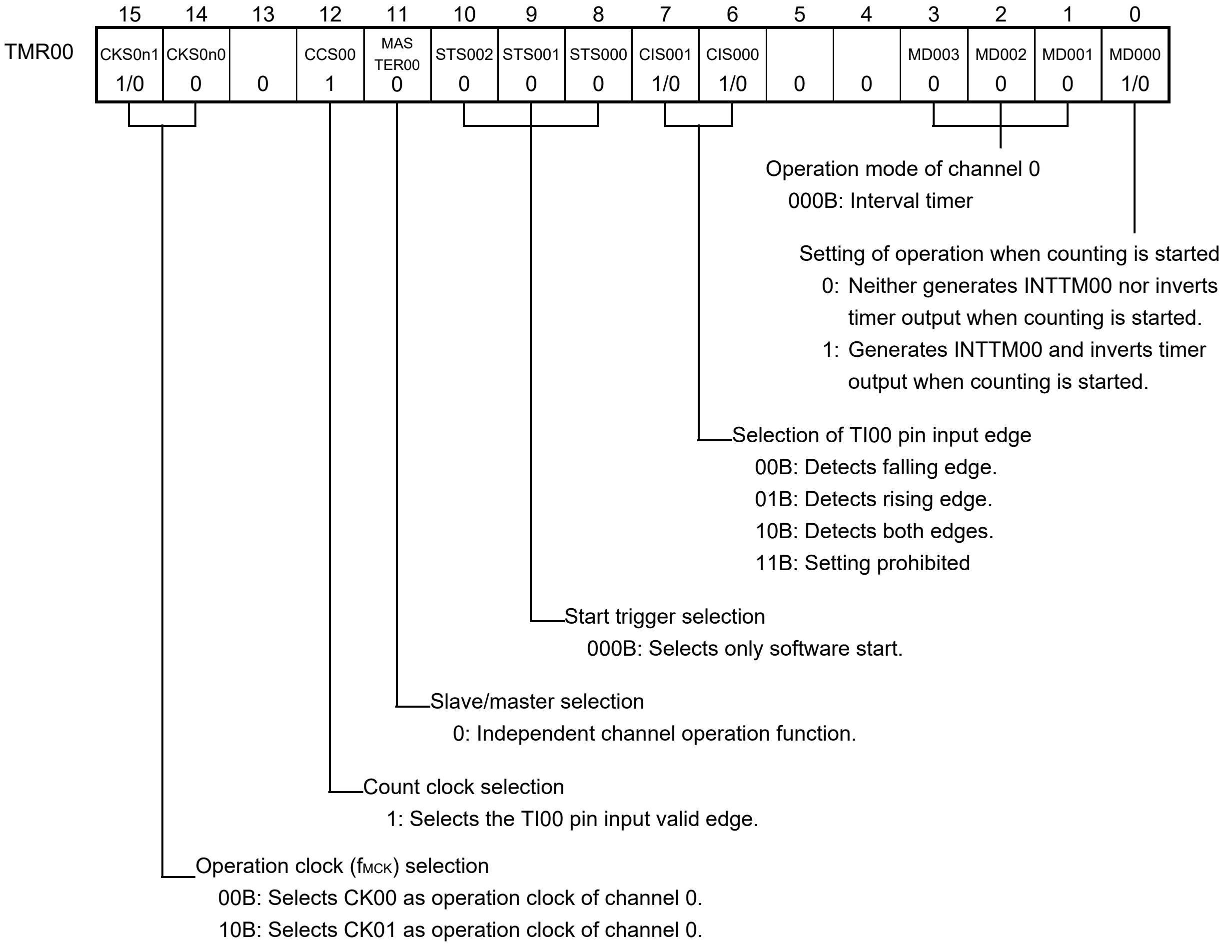
Figure 6-52. Example of Basic Timing of Operation as Frequency Divider (MD000 = 1)



Remark		
	TS00:	Bit n of timer channel start register 0 (TS0)
	TE00:	Bit n of timer channel enable status register 0 (TE0)
	TI00:	TI00 pin input signal
	TCR00:	Timer count register 00 (TCR00)
	TDR00:	Timer data register 00 (TDR00)
	TO00:	TO00 pin output signal

Figure 6-53. Example of Set Contents of Registers During Operation as Frequency Divider

(a) Timer mode register 00 (TMR00)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR00	CKS0n1 1/0	CKS0n0 0	0	CCS00 1	MAS TER00 0	STS002 0	STS001 0	STS000 0	CIS001 1/0	CIS000 1/0	0	0	MD003 0	MD002 0	MD001 0	MD000 1/0

Operation mode of channel 0
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTM00 nor inverts timer output when counting is started.
1: Generates INTTM00 and inverts timer output when counting is started.

Selection of TI00 pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Independent channel operation function.

Count clock selection
1: Selects the TI00 pin input valid edge.

Operation clock (f_{MCK}) selection
00B: Selects CK00 as operation clock of channel 0.
10B: Selects CK01 as operation clock of channel 0.

(b) Timer output register 0 (TO0)

	Bit 0
TO0	TO00 1/0

0: Outputs 0 from TO00.
1: Outputs 1 from TO00.

(c) Timer output enable register 0 (TOE0)

	Bit 0
TOE0	TOE00 1/0

0: Stops the TO00 output operation by counting operation.
1: Enables the TO00 output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit 0
TOL0	TOL00 0

0: Cleared to 0 when master channel output mode (TOM00 = 0)

(e) Timer output mode register 0 (TOM0)

	Bit 0
TOM0	TOM00 0

0: Sets master channel output mode.

Figure 6-54. Operation Procedure When Frequency Divider Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel and selects the detection edge). Sets interval (period) value to timer data register 00 (TDR00).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOM00 bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL00 bit to 0.	The TO00 pin goes into Hi-Z output state.
	Sets the TO00 bit and determines default level of the TO00 output.	The TO00 default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE00 bit to 1 and enables operation of TO00.	TO00 does not change because channel stops operating.
	Clears the port register and port mode register to 0.	The TO00 pin outputs the TO00 set level.
Operation start	Sets the TOE00 bit to 1 (only when operation is resumed). Sets the TS00 bit to 1. The TS00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 1, and count operation starts. Value of the TDR00 register is loaded to timer count register 00 (TCR00) at the count clock input. INTTM00 is generated and TO00 performs toggle operation if the MD000 bit of the TMR00 register is 1.
During operation	Set value of the TDR00 register can be changed. Sets corresponding bit of noise filter enable register 1, 2 (NFEN1, NFEN2) to 1. The TCR00 register can always be read. The TSR00 register is not used. Set values of the TO0 and TOE0 registers can be changed. Set values of the TMR00 register, TOM00, and TOL00 bits cannot be changed.	Counter (TCR00) counts down. When count value reaches 0000H, the value of the TDR00 register is loaded to the TCR00 register again, and the count operation is continued. By detecting TCR00 = 0000H, INTTM00 is generated and TO00 performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT00 bit is set to 1. The TT00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 0, and count operation stops. The TCR00 register holds count value and stops. The TO00 output is not initialized but holds current status.
	The TOE00 bit is cleared to 0 and value is set to the TO00 bit.	The TO00 pin outputs the TO00 set level.
TAU stop	To hold the TO00 pin output level Clears the TO00 bit to 0 after the value to be held is set to the port register. When holding the TO00 pin output level is not necessary Setting not required.	The TO00 pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO00 bit is cleared to 0 and the TO00 pin is set to port mode).

Operation is resumed.

6.8.4 Operation as input pulse interval measurement

The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured.

In addition, the count value can be captured by using software operation (TSmn = 1) as a capture trigger while the TEmn bit is set to 1. The pulse interval can be calculated by the following expression.

TImn input pulse interval = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error of up to one operating clock cycle occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TCRmn register counts up from 0000H in synchronization with the count clock.

When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

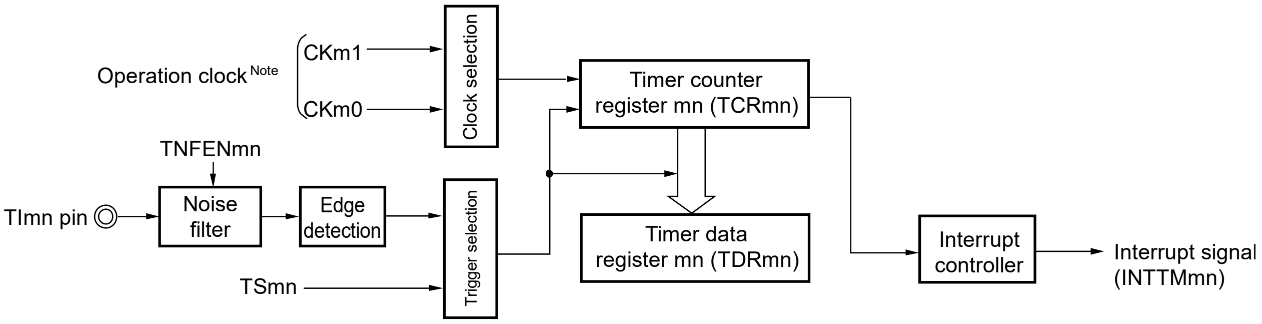
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Set the STSmn2 to STSmn0 bits of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

When TEmn = 1, a software operation (TSmn = 1) can be used as a capture trigger, instead of using the TImn pin input.

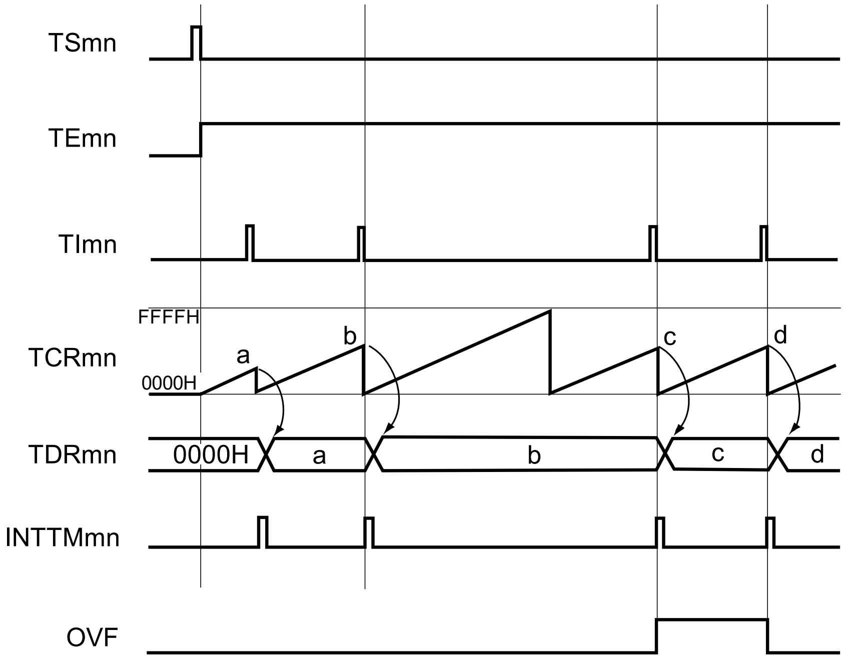
Figure 6-55. Block Diagram of Operation as Input Pulse Interval Measurement



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-56. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)



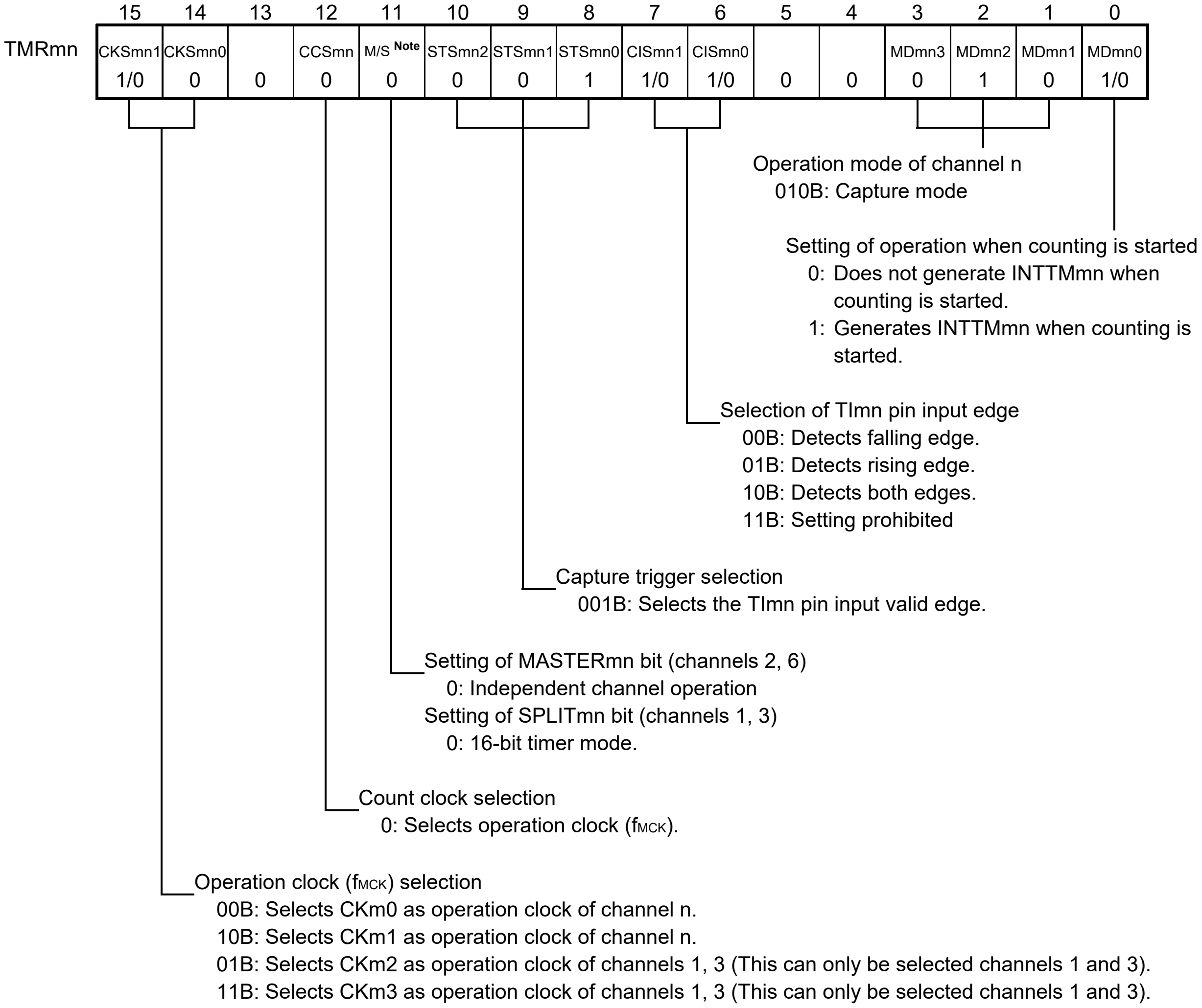
Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

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 6-57. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	0	0	0	1	1/0	1/0	0	0	0	1	0	1/0

Operation mode of channel n
010B: Capture mode

Setting of operation when counting is started
0: Does not generate INTTMmn when counting is started.
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Capture trigger selection
001B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channels 2, 6)
0: Independent channel operation
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-58. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) 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 TImn pin input valid edge is detected, the count value is transferred (captured) to timer data register mn (TDRmn). At the same time, the TCRmn register is cleared to 0000H, and the INTTMmn signal is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

6.8.5 Operation as input signal high-/low-level width measurement

By starting counting at one edge of the TImn pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TImn can be measured. The signal width of TImn can be calculated by the following expression.

Signal width of TImn input = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error equivalent to one operation clock occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture & one-count mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TEmn bit is set to 1 and the TImn pin start edge detection wait status is set.

When the TImn pin input start edge (rising edge of the TImn pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TImn pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register mn (TDRmn) and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops at the value “value transferred to the TDRmn register + 1”, and the TImn pin start edge detection wait status is set. After that, the above operation is repeated.

As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

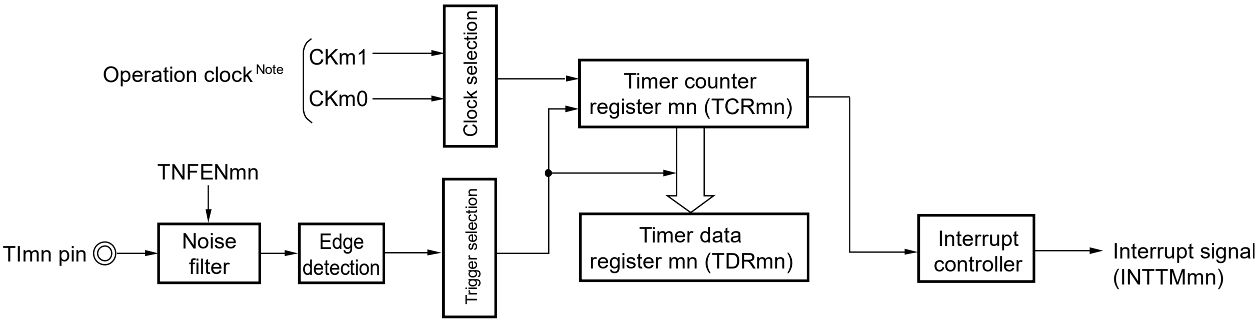
If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.

Because this function is used to measure the signal width of the TImn pin input, the TSmn bit cannot be set to 1 while the TEmn bit is 1.

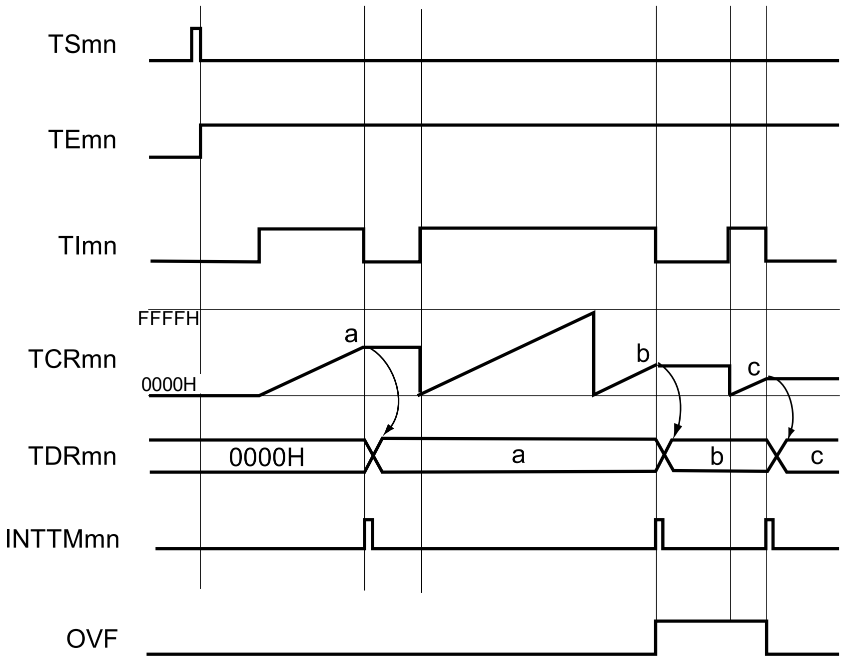
CISmn1, CISmn0 of TMRmn register = 10B: Low-level width is measured.
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 6-59. 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 6-60. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

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 6-61. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	0	0	1	0	1	1/0	0	0	1	1	0	0

Operation mode of channel n
110B: Capture & one-count

Setting of operation when counting is started
0: Does not generate INTTMmn when counting is started.

Selection of TImn pin input edge
10B: Both edges (to measure low-level width)
11B: Both edges (to measure high-level width)

Start trigger selection
010B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channels 2, 6)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
1: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-62. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the TImn pin start edge detection wait status is set.
	Detects the TImn pin input count start valid edge.	Clears timer count register mn (TCRmn) to 0000H and starts counting up.
During operation	The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to timer data register mn (TDRmn) and INTTMmn is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. The TCRmn register stops the count operation until the next TImn pin start edge is detected.
Operation stop	The TTmn bit is set to 1. The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

6.8.6 Operation as delay counter

It is possible to start counting down when the valid edge of the TImn pin input is detected (an external event), and then generate INTTMmn (a timer interrupt) after any specified interval.

It can also generate INTTMmn (timer interrupt) at any interval by making a software set TSmn = 1 and the count down start during the period of TEmn = 1.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

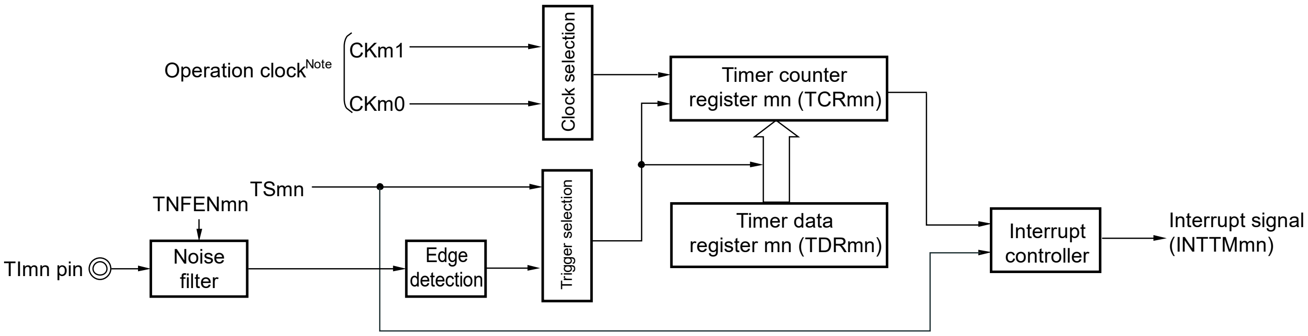
Timer count register mn (TCRmn) operates as a down counter in the one-count mode.

When the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1, the TEmn, TEHm1, TEHm3 bits are set to 1 and the TImn pin input valid edge detection wait status is set.

Timer count register mn (TCRmn) starts operating upon TImn pin input valid edge detection and loads the value of timer data register mn (TDRmn). The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next TImn pin input valid edge is detected.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

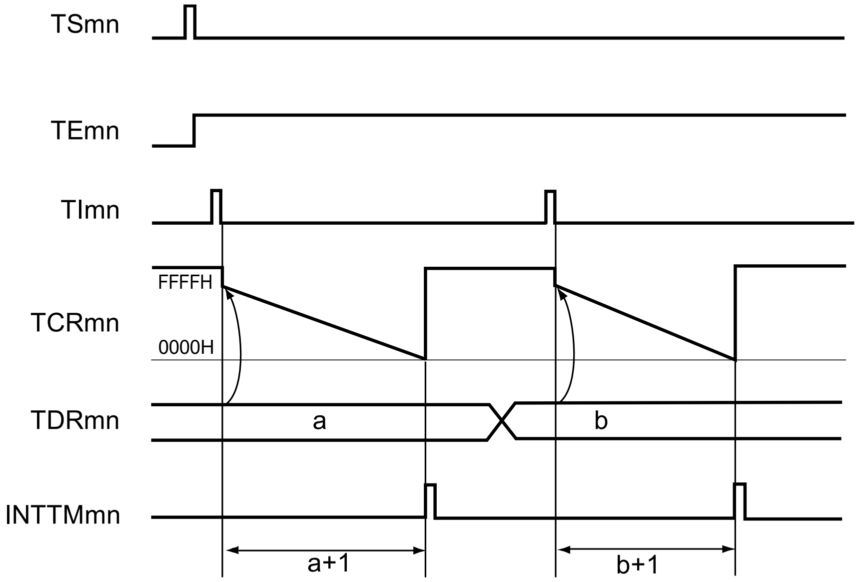
Figure 6-63. Block Diagram of Operation as Delay Counter



Note For using channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-64. Example of Basic Timing of Operation as Delay Counter



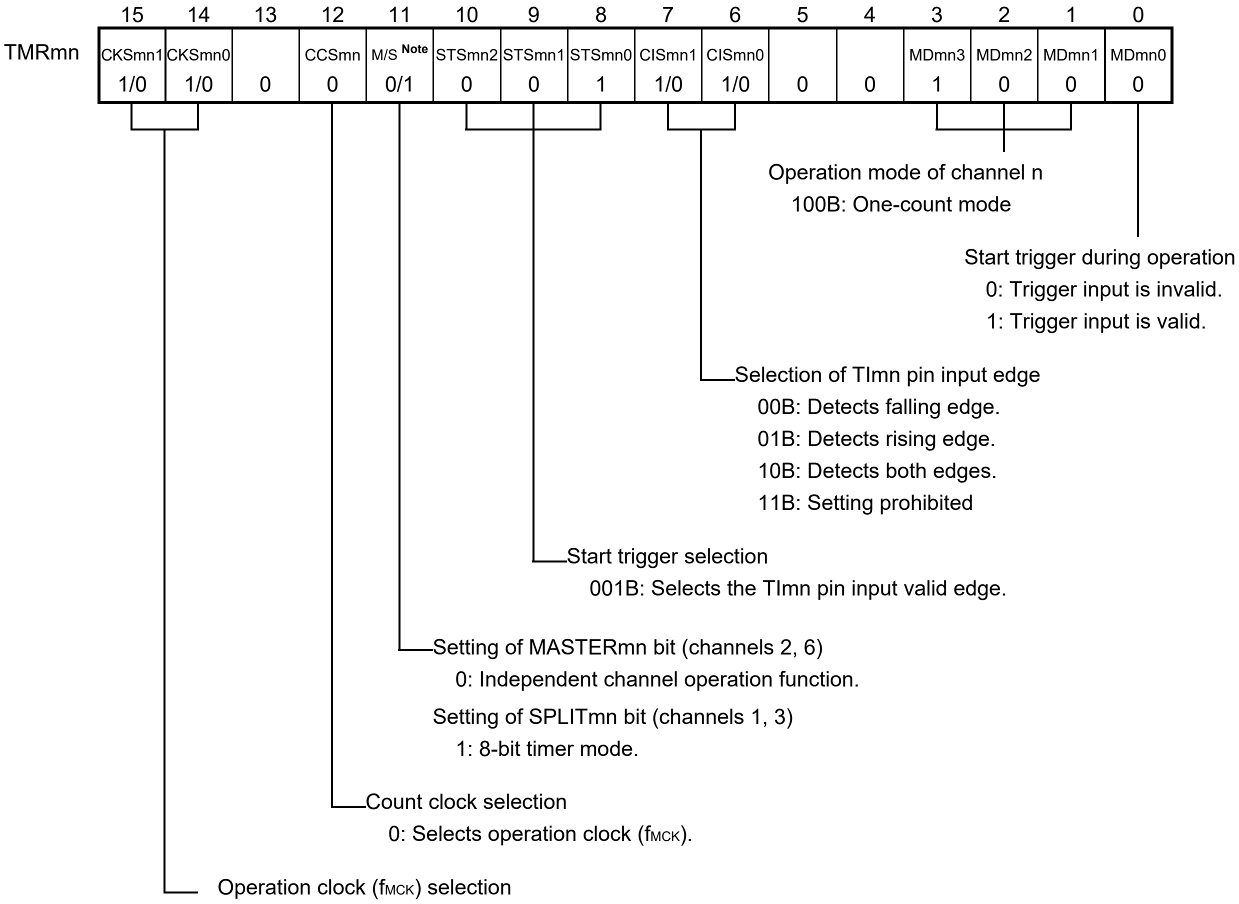
Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

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 6-65. Example of Set Contents of Registers to Delay Counter (1/2)

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0/1	0	0	1	1/0	1/0	0	0	1	0	0	0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.
1: Trigger input is valid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channels 2, 6)
0: Independent channel operation function.

Setting of SPLITmn bit (channels 1, 3)
1: 8-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops the TOmn output operation by counting operation.

Note	
TMRm2, TMRm6:	MASTERmn bit
TMRm1, TMRm3:	SPLITmn bit
TMRm0, TMRm7:	Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-65. Example of Set Contents of Registers to Delay Counter (2/2)

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 7)

Figure 6-66. Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 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 TImn pin input valid edge detection 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 TCRmn counts down to 0000H, INTTMmn is output, and counting stops (which leaves TCRmn at 0000H) until the next TImn pin input.
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 TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3, 6, 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.

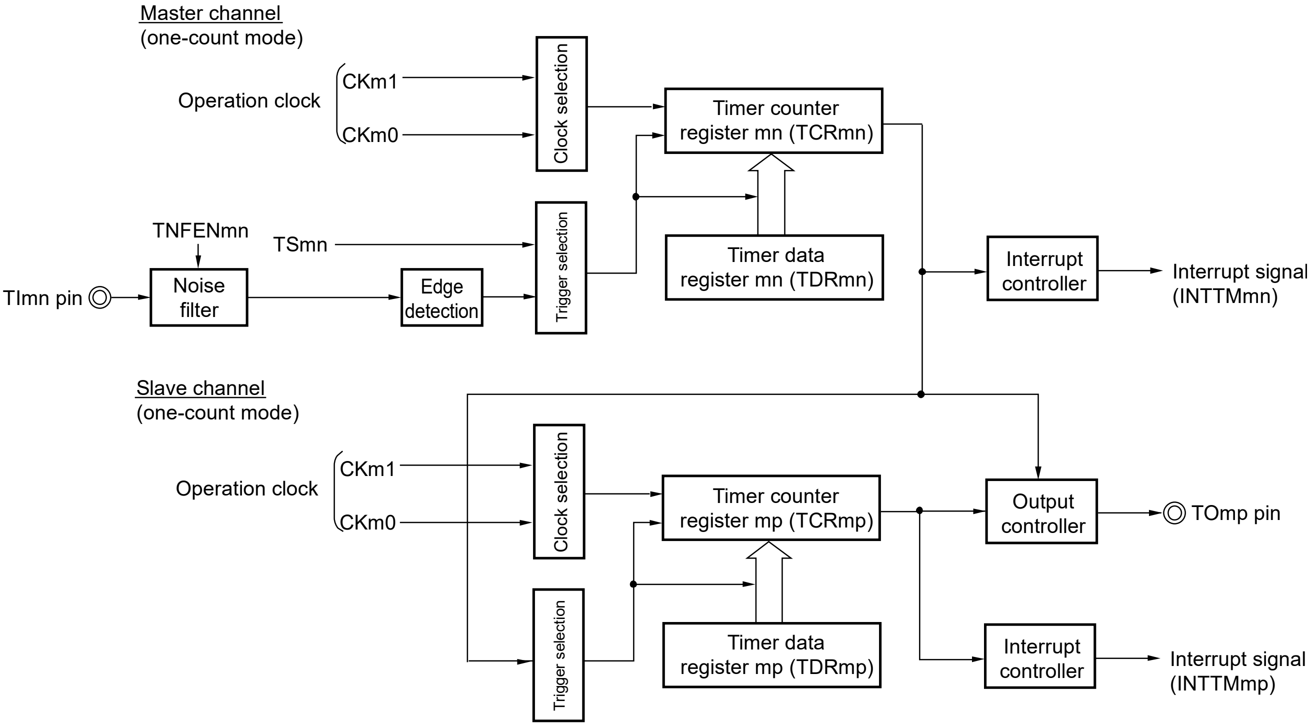
The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register 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 operation, 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, 6)
p: Slave channel number (n < p ≤ 7)

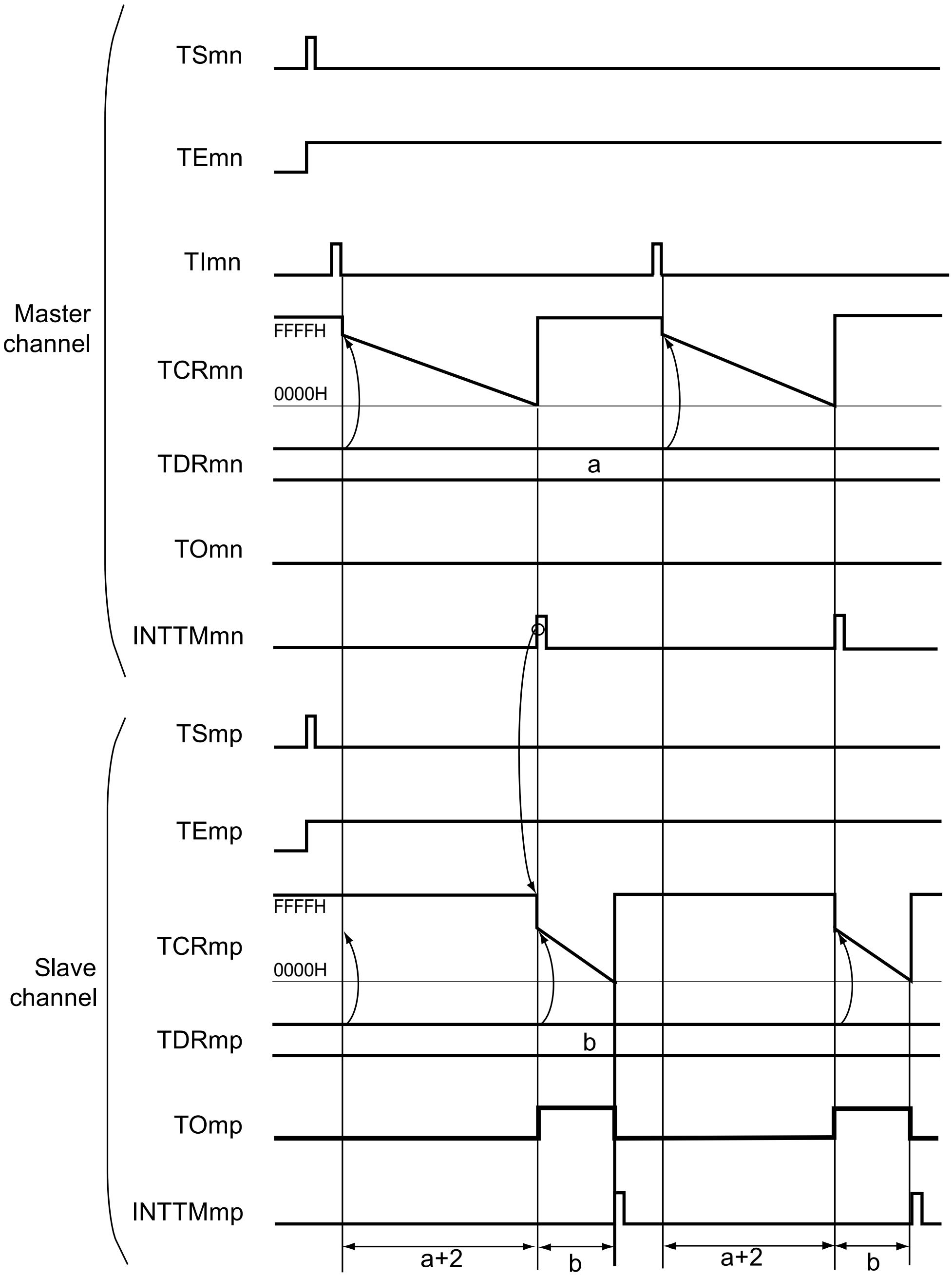
Figure 6-67. Block Diagram of Operation as One-Shot Pulse Output Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 6)

p: Slave channel number (n < p ≤ 7)

Figure 6-68. Example of Basic Timing of Operation as One-Shot Pulse Output Function



Remarks **1.** m: Unit number (m = 0), n: Channel number (n = 0, 2, 6)

p: Slave channel number ($n < p \leq 7$)

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 6-69. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	1	1/0	1/0	0	0	1	0	0	0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TImn pin input valid edge.

Slave/master selection
1: Master channel.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channels n.
10B: Selects CKm1 as operation clock of channels n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

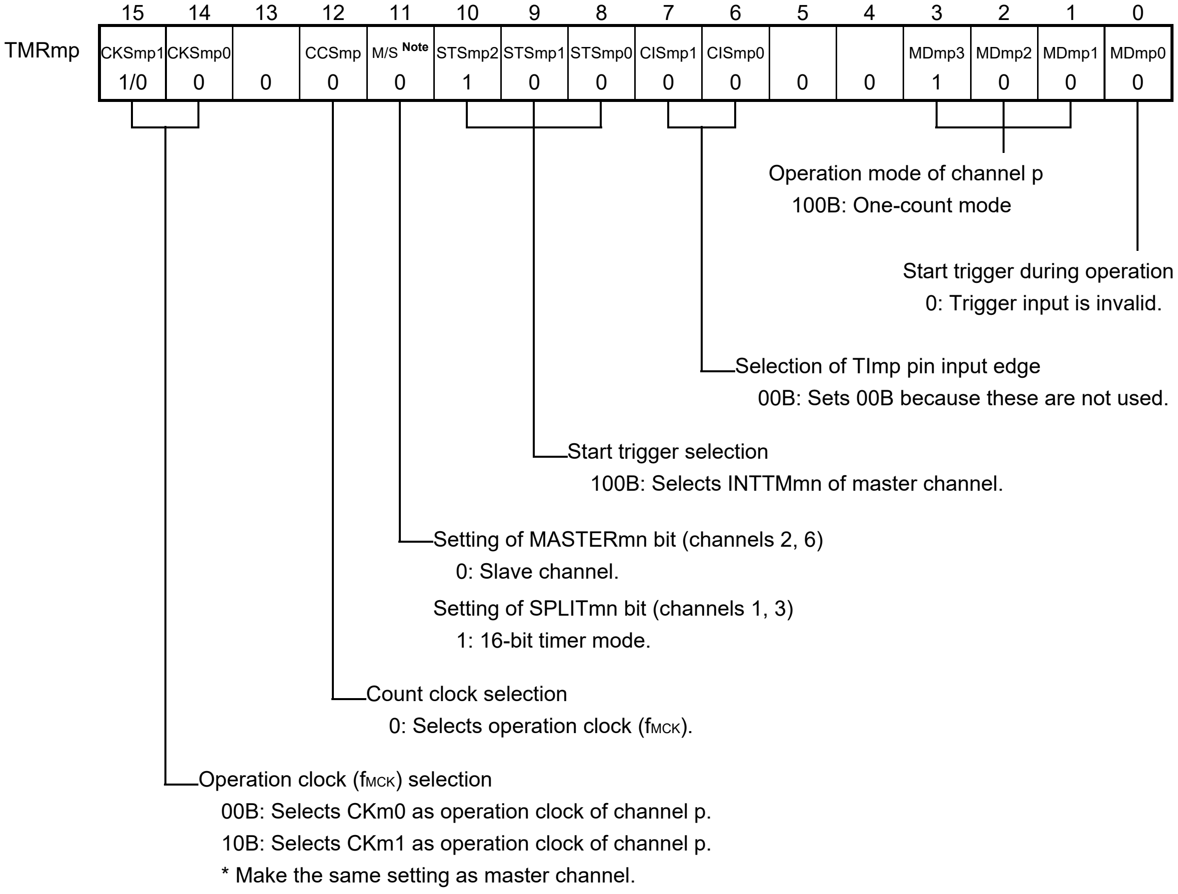
	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 6)

Figure 6-70. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register mp (TMRmp)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

Operation mode of channel p
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmn bit (channels 2, 6)
0: Slave channel.

Setting of SPLITmn bit (channels 1, 3)
1: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p.
10B: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p	
TOm	TOmp 1/0	0: Outputs 0 from TOmp. 1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p	
TOEm	TOEmp 1/0	0: Stops the TOmp output operation by counting operation. 1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p	
TOLm	TOLmp 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p	
TOMm	TOMmp 1	1: Sets the slave channel output mode.

Note		
	TMRm2, TMRm6:	MASTERmn bit
	TMRm1, TMRm3:	SPLITmp bit
	TMRm7:	Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-71. Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable registers 0 (PER0) to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 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 0 (off) or 1 (on). 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 6-71. Operation Procedure of One-Shot Pulse Output Function (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	The TEmn and TEmp bits are set to 1 and the master channel enters the TImn input edge detection wait status. Counter stops operating.
	Count operation of the master channel is started by start trigger detection of the master channel. ● Detects the TImn pin input valid edge. ● Sets the TSmn bit of the master channel to 1 by software Note. **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. Sets corresponding bit of noise filter enable register 1 (NFEN1) to 1. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers by slave channel can be changed.	Master channel loads the value of the TDRmn register to timer count register mn (TCRmn) when the TImn pin valid input edge is detected, and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and the counter stops until the next valid edge is input to the TImn pin. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit.	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The 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, 6)
p: Slave channel number ($n < p \leq 7$)

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.

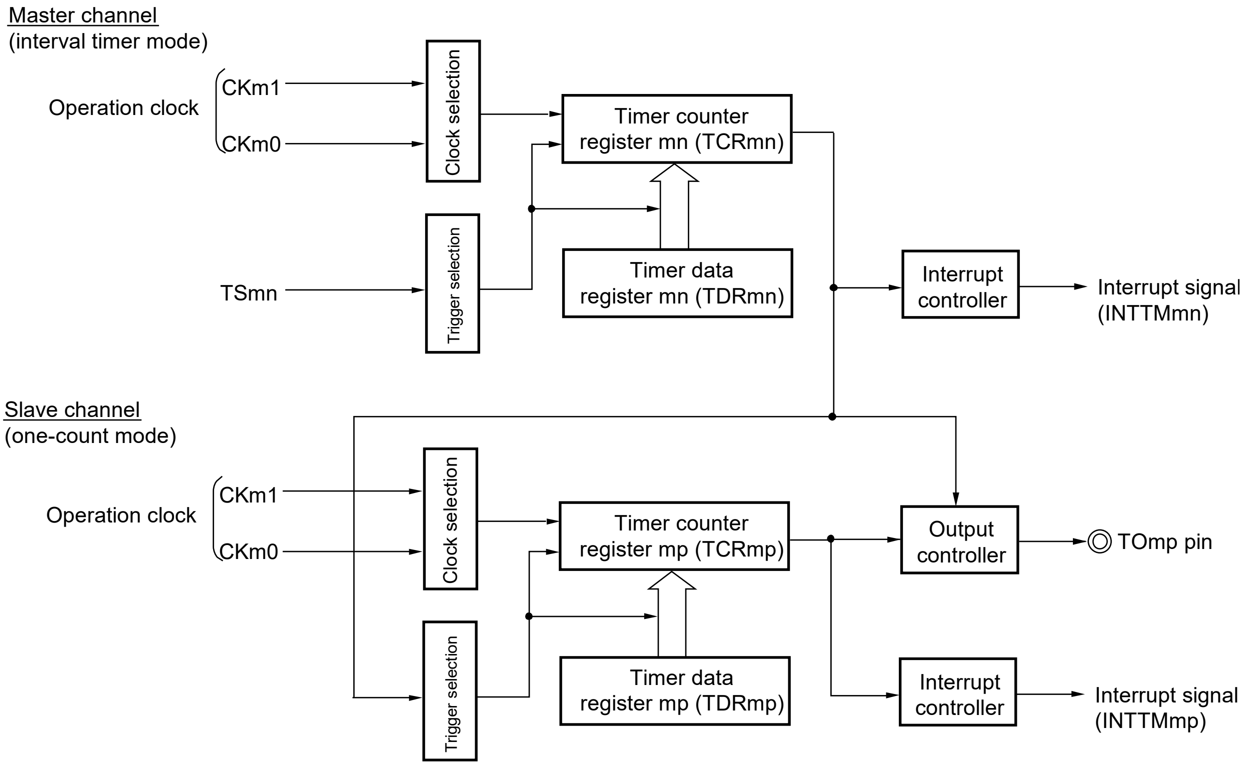
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.

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.

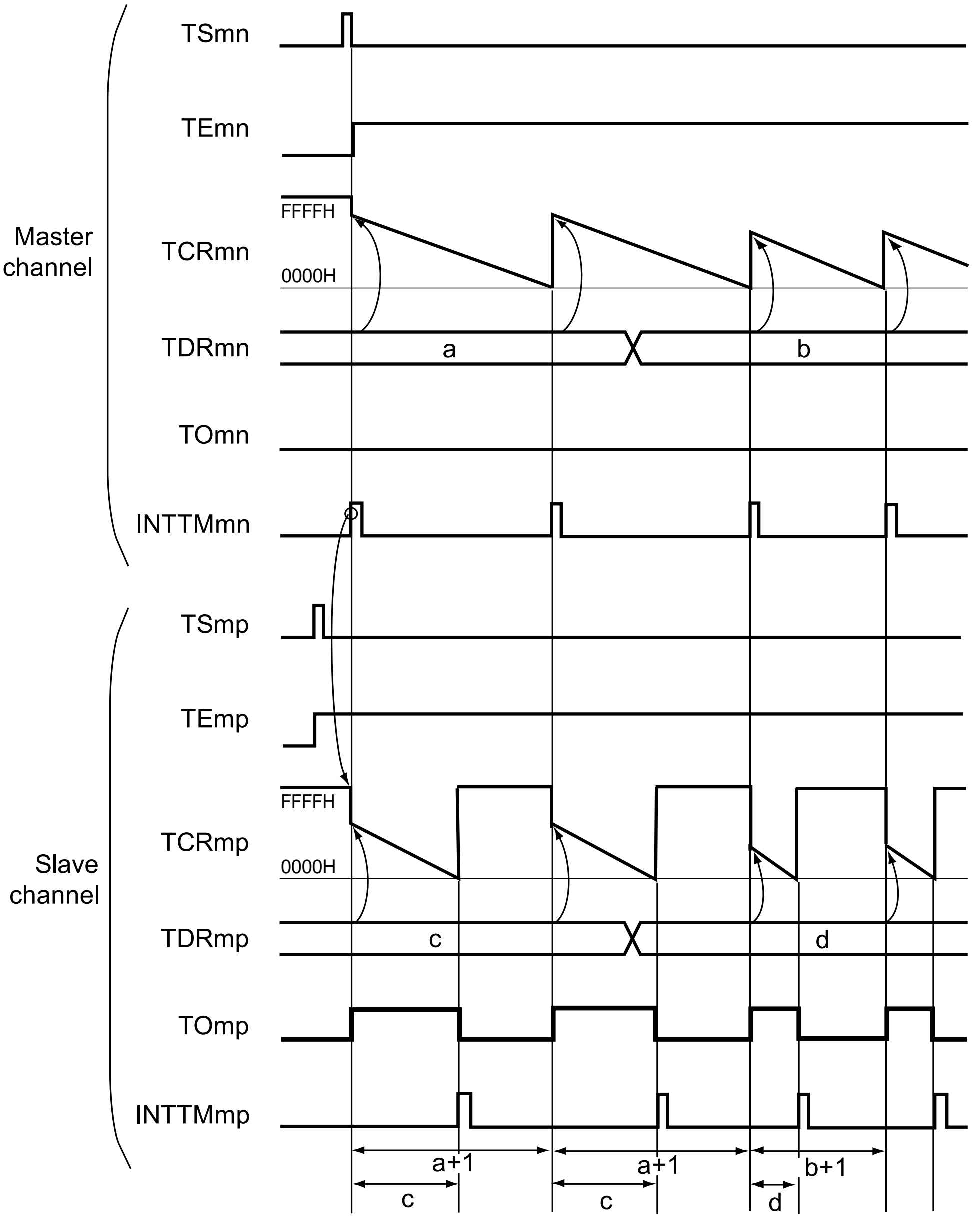
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-72. Block Diagram of Operation as PWM Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 6)
p: Slave channel number ($n < p \leq 7$)

Figure 6-73. Example of Basic Timing of Operation as PWM Function



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 2, 6)

p: Slave channel number (n < p ≤ 7)

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)

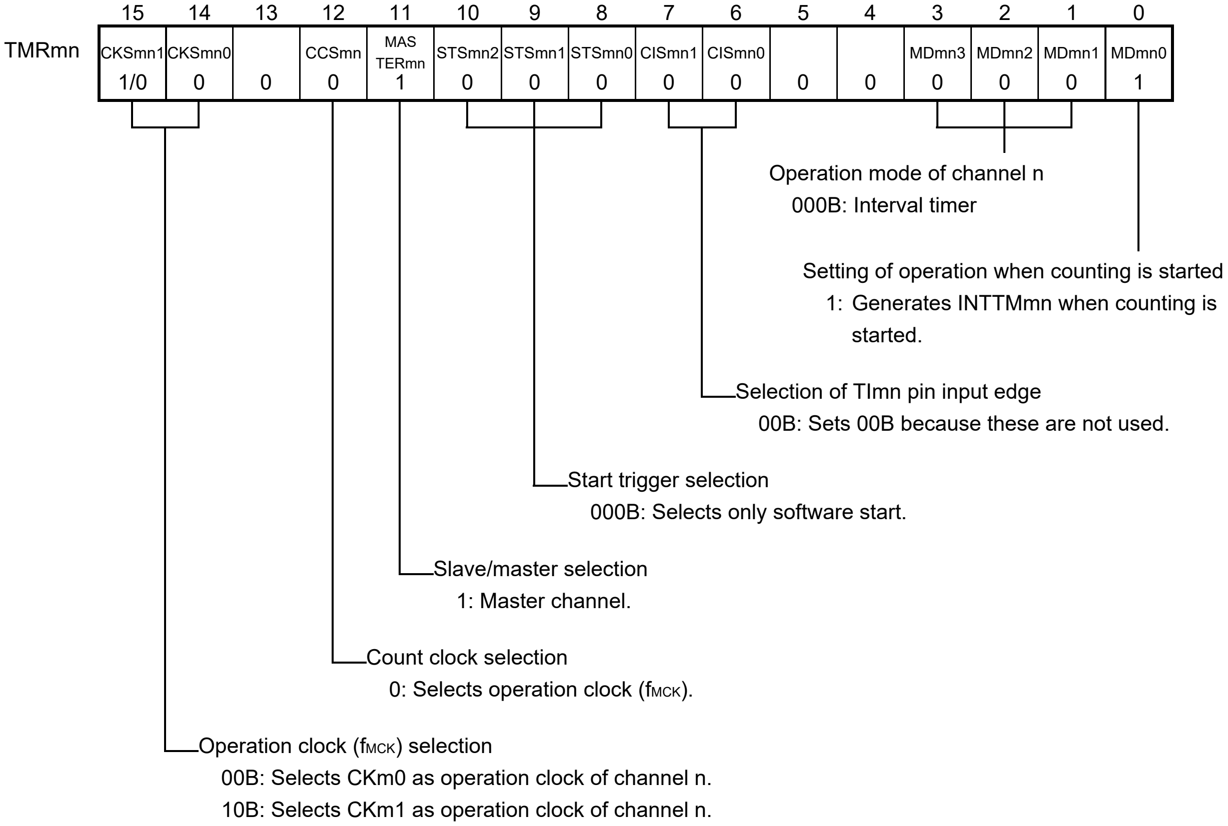
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)

TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)

TOmn, TOmp: TOmn and TOmp pins output signal

Figure 6-74. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)



Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Master channel.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

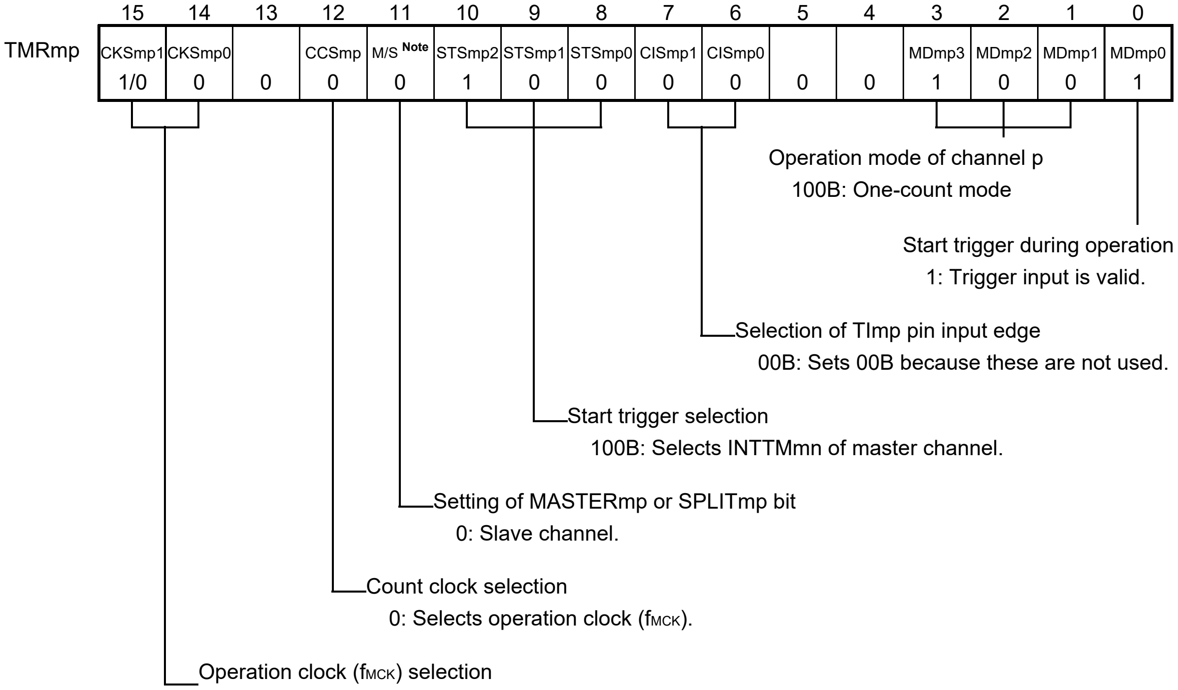
(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 6)

Figure 6-75. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register mp (TMRmp)



Operation mode of channel p
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

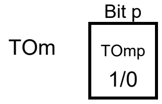
Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmp or SPLITmp bit
0: Slave channel.

Count clock selection
0: Selects operation clock (f_{MCK}).

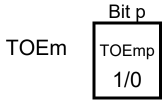
Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p.
10B: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)



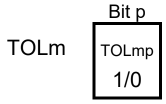
0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)



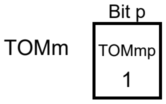
0: Stops the TOmp output operation by counting operation.
1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)



0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)



1: Sets the slave channel output mode.

Note TMRm7: Fixed to 0
TMRm1, TMRm3: SPLITmp bit

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2, 6)
p: Slave channel number ($n < p \le 7$)

Figure 6-76. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit.	The TOmp pin goes into Hi-Z output state.
	Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

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

Figure 6-76. Operation Procedure When PWM Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn), and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel, the value of the TDRmp register is loaded to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit.	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The 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, 6)
p: Slave channel number ($n < p \leq 7$)

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.

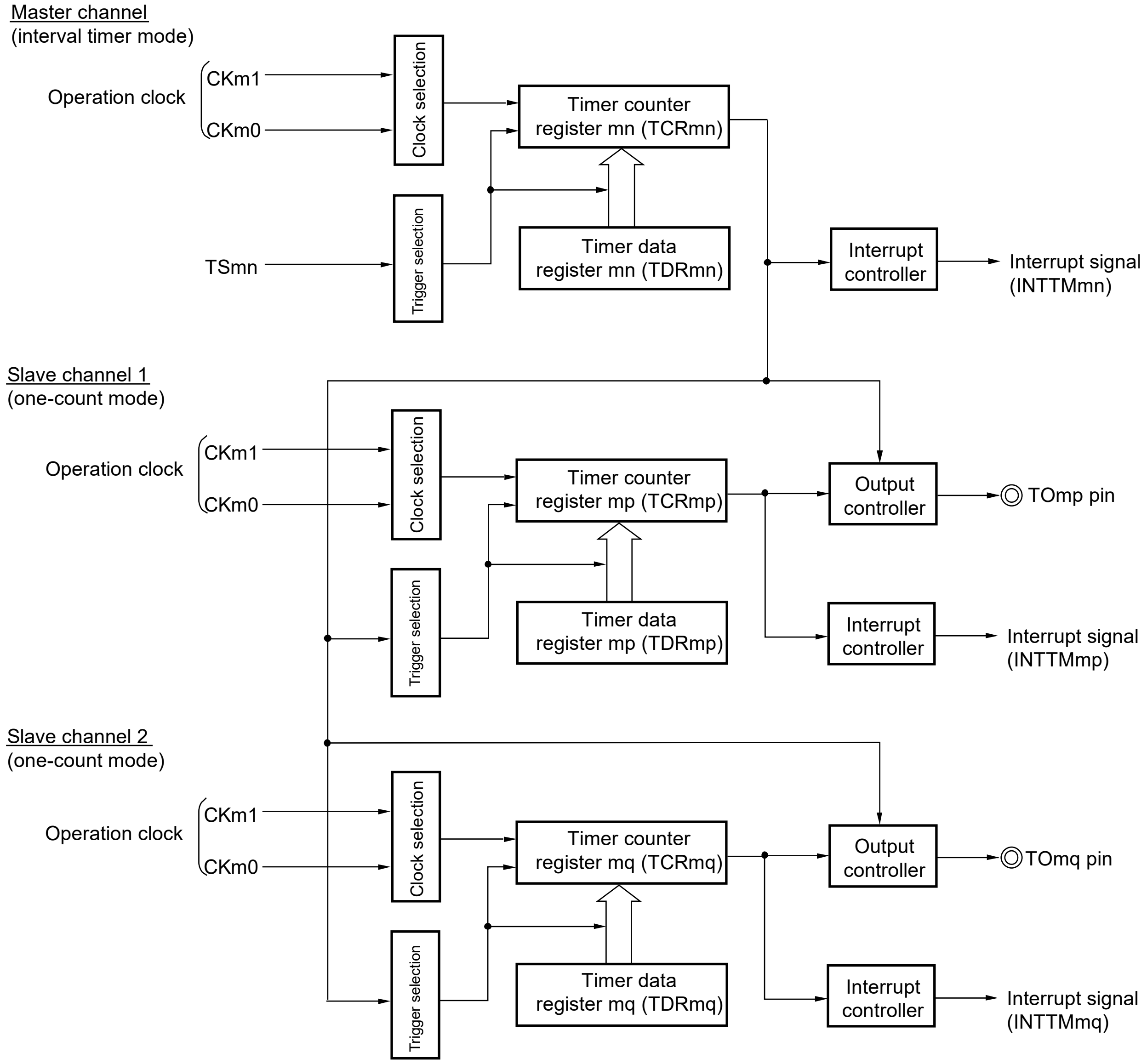
In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a 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 seven 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).

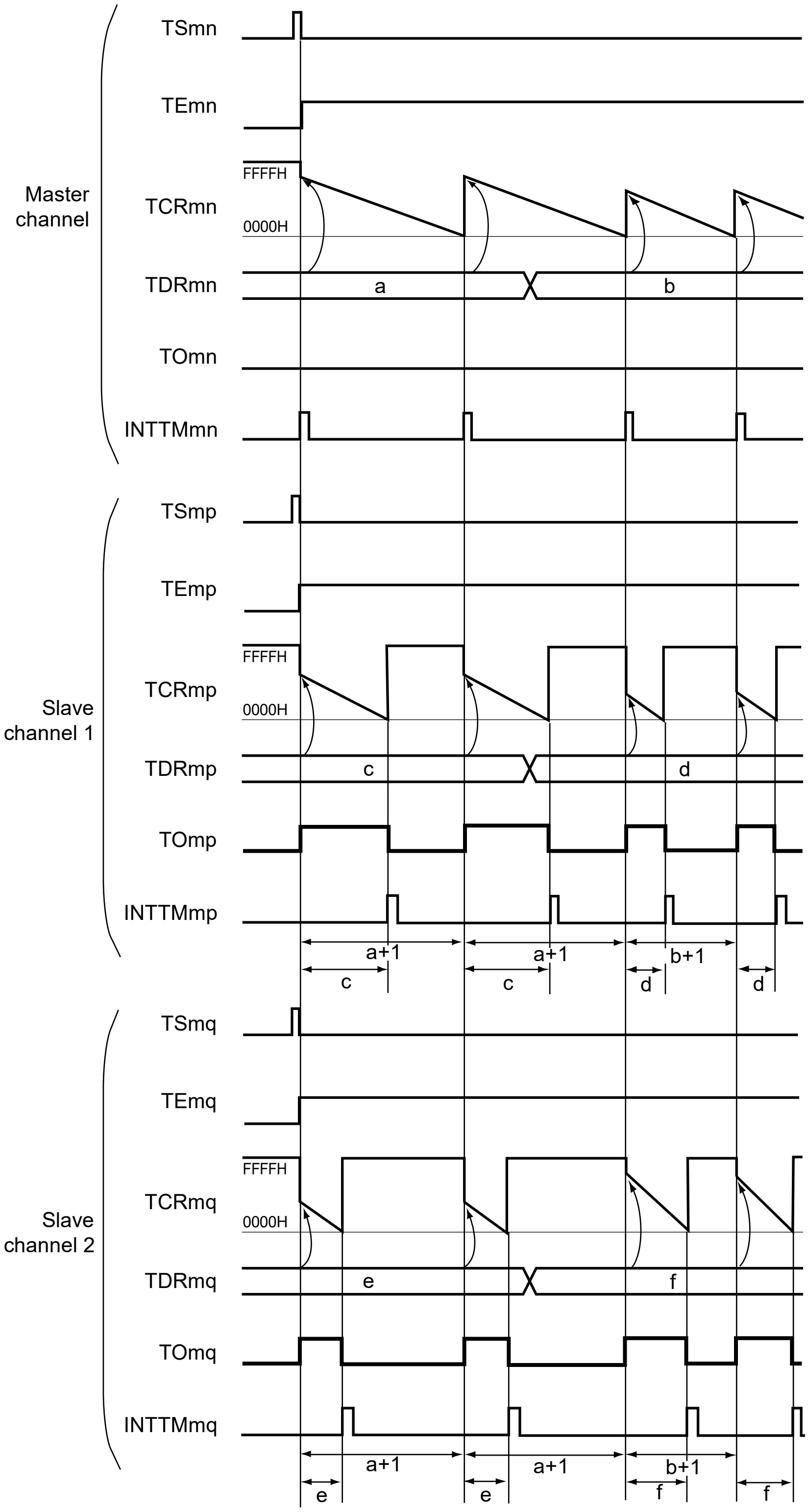
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \leq 7$ (Where p and q are integers greater than n)

Figure 6-77. 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)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6-78. Example of Basic Timing of Operation as Multiple PWM Output Function (Output two types of PWMs)



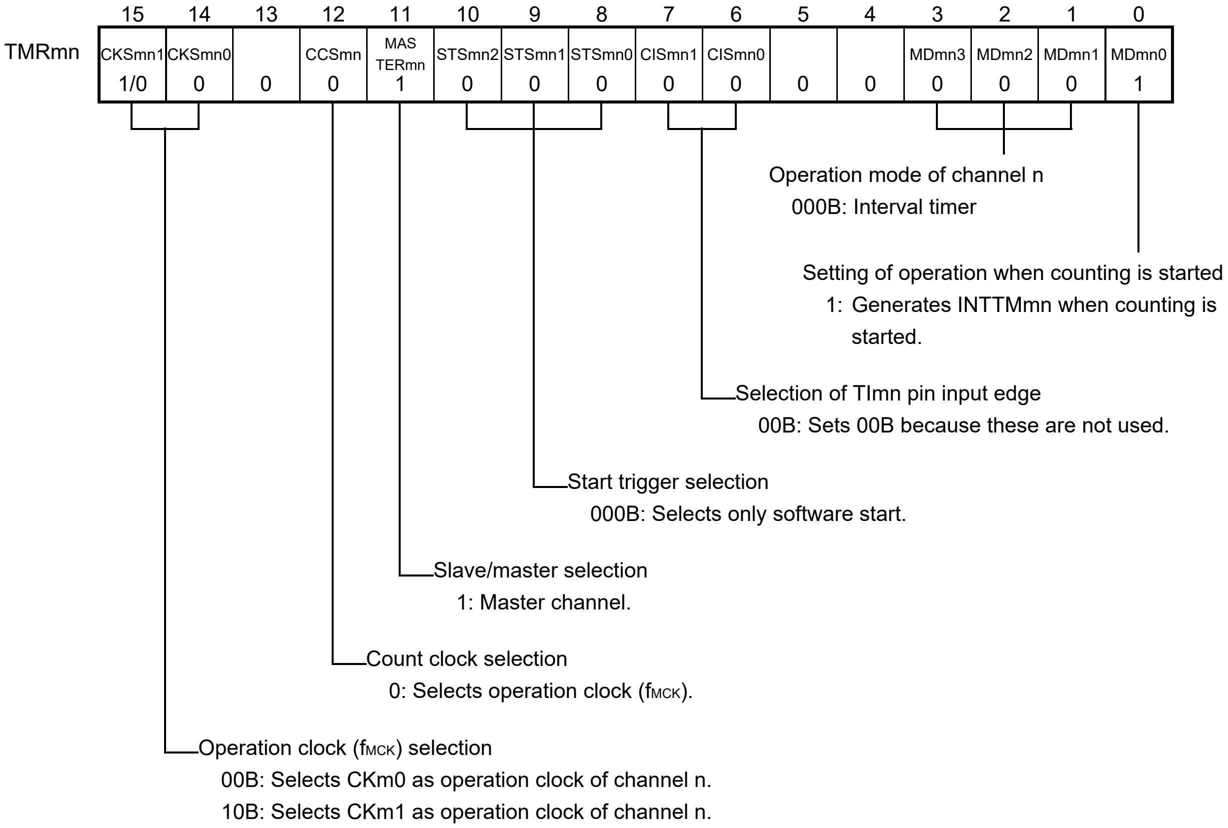
(**Remarks** are listed on the next page.)

Remarks1. m: Unit number ($m = 0$), n: Channel number ($n = 0, 2$)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \le 7$ (Where p and q are integers greater than n)

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 6-79. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Master channel.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

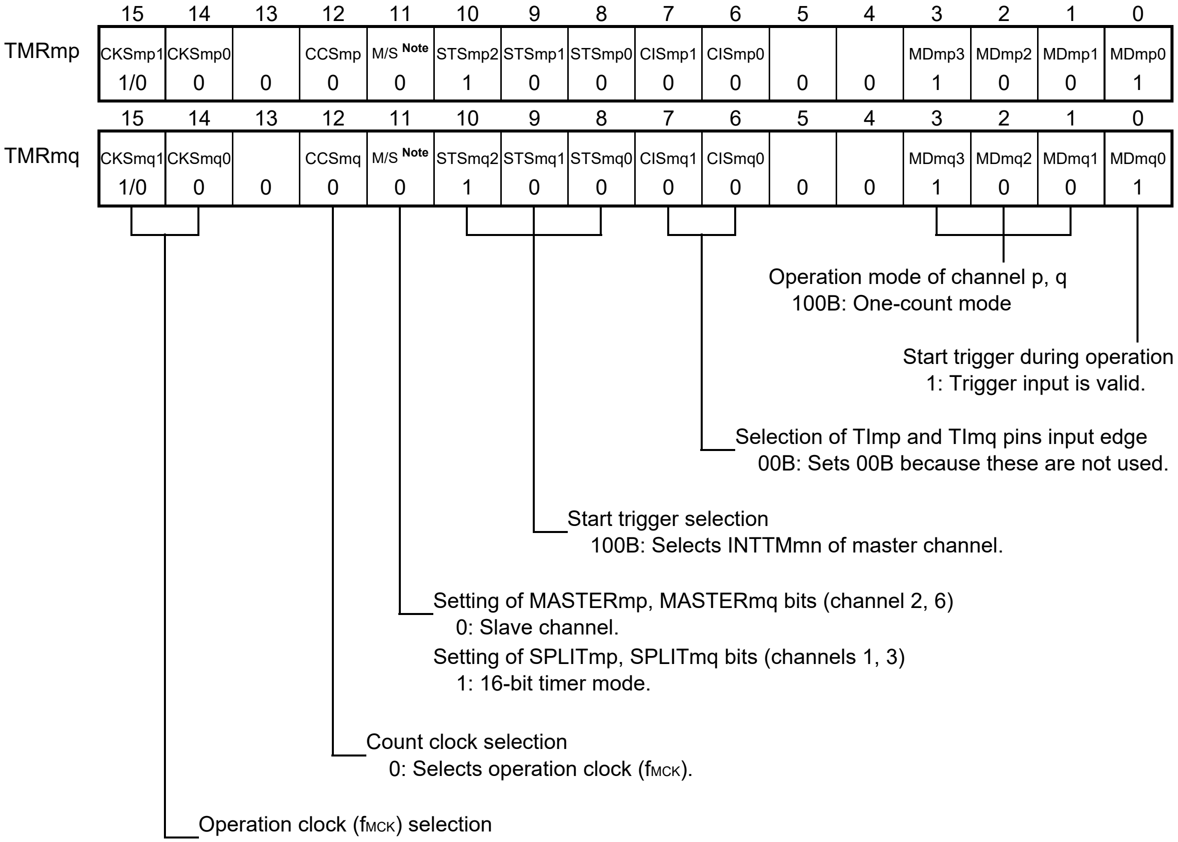
	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

Figure 6-80. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)

(a) Timer mode register mp, mq (TMRmp, TMRmq)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmq	CKSmq1	CKSmq0		CCSmq	M/S Note	STSmq2	STSmq1	STSmq0	CISmq1	CISmq0			MDmq3	MDmq2	MDmq1	MDmq0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p, q
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp and TImq pins input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmp, MASTERmq bits (channel 2, 6)
0: Slave channel.
Setting of SPLITmp, SPLITmq bits (channels 1, 3)
1: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p, q.
10B: Selects CKm1 as operation clock of channel p, q.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit q	Bit p
TOm	TOmq	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	
TMRm7:	Fixed to 0
TMRm1, TMRm3:	SPLITmp, SPLIT0q bit
TMRm2, TMRm6:	MASTERmp, MASTERmq bit

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \leq 7$ (Where p and q are integers greater than n)

Figure 6-81. Operation Procedure When Multiple PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp, 0q (TMRmn, TMRmp, TMRmq) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp and TDRmq registers of the slave channels.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOMmp and TOMmq bits of timer output mode register m (TOMm) are set to 1 (slave channel output mode). Sets the TOLmp and TOLmq bits.	The TOmp and TOmq pins go into Hi-Z output state.
	Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. →	The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOmq. →	TOmp and TOmq do not change because channels stop operating.
	Clears the port register and port mode register to 0. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.

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

Figure 6-81. Operation Procedure When Multiple PWM Output Function Is Used (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 TSR0q 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 TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The 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)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 7 (Where p and q are a 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 6 and 7 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 7 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 6-82. Remote Control Output

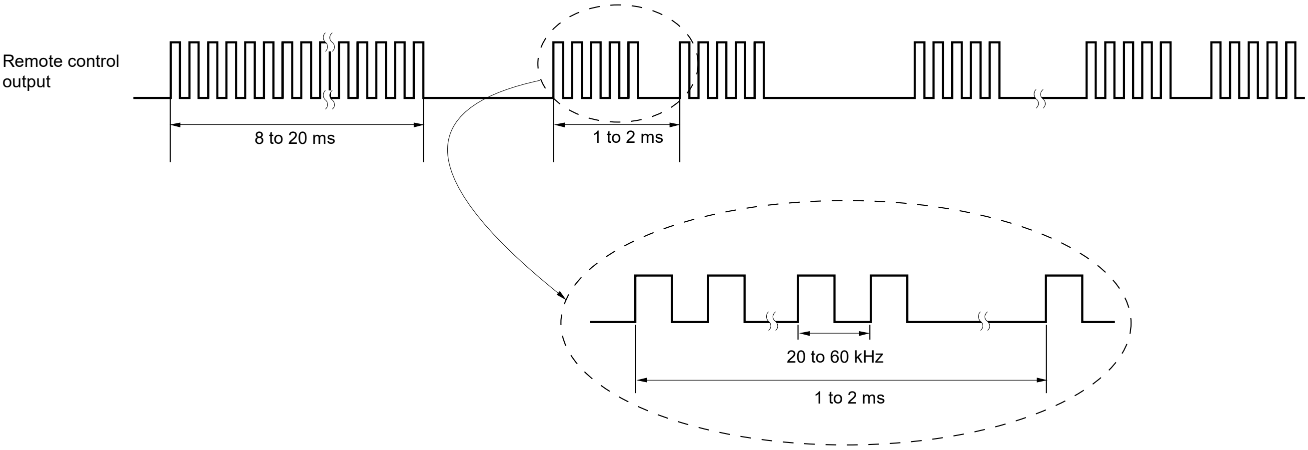


Figure 6-83 shows the steps for setting the remote control output.

Figure 6-83. Procedure for Setting Remote Control Output (1/2)

	Software Operation	Hardware Status
Pin mode setting	Sets the PFSEG17 bit of PFSEG2 register, PM32 bit of PM3 register, PU3 bit of PFSEG2 register and P32 bit of P3 register to 1	Remote control output is invalid P32/TO03 pin is low-level output
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Remote control output setting	The TOS0 bit of the Timer output select register (TOS) is set to 1.	Remote control output is valid The P32/TO03 pin outputs the result (Low) of ANDing TO03 (Low) and TO07 (Low). P32/TO03 pin can only be used as a remote control output P53/TO07 pin can only be used as a alternative function other than timer output
Channel default setting	Sets timer mode register mn (TMRmn) to 0801H and sets timer mode register mp (TMRmp) to 0409H determines operation mode of channels).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets master channels. The TOMmn bit of timer output mode register m (TOMm) is set to 0 (master channel output mode). Sets the TOLmn bit.	The TOmn pin goes into Hi-Z output state.
	Sets the TOmn bit and determines default level of the TOmn output.	The TOmn default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmn bit to 1 and enables operation of TOmn.	TOmn does not change because channels stop operating.
	Clears the port register and port mode register to 0.	The TOmn pin outputs the TOmn set level.
	Sets slave channels. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit.	The TOmp pin goes into Hi-Z output state.
	Sets the TOmp bit and determines default level of the TOmp output.	The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp.	TOmp does not change because channels stop operating.
	Clears the port register and port mode register to 0.	The TOmp pin outputs the TOmp set level.

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

Figure 6-83. Procedure for Setting Remote Control Output (2/2)

	Software Operation	Hardware Status
Operation start	The cycle of the mask waveform (start code) and its high-level width are set. TDR02 = The cycle of the mask waveform - 1 TDR03 = High-level width of the mask waveform The cycle of the carrier waveform and its high-level width are set. TDR06 = The cycle of the carrier waveform - 1 TDR07 = High-level width of the carrier waveform The TSmn bit (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, TEmq = 1 When the master channel starts counting, INTTM02 is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	The setting of the TMRmn and TMRmp registers, and the TOMmn, TOMmp, TOLmn, and TOLmp bits must not be changed. The TCRmn and TCRmp registers can always be read. Wait for an interrupt signal (INTTM02) Last code bit? ● If it is not the end code bit, the cycle and high-level width of the next mask waveform are specified. TDR02 = The cycle of the mask waveform - 1 TDR03 = High-level width of the mask waveform **Caution Setting must finish before the TCR02 value reaches 0000H.** ● If it is the end code bit, the operation stops	TO03 outputs the mask waveform and TO07 outputs the carrier waveform in accordance with the settings of the cycle and high-level width. The P32/TO03 pin outputs the result of ANDing the TO03 and TO07 outputs (a remote control output (carrier waveform) until TCR03 reaches 0000H; a low-level remote control output until TCR02 reaches 0000H and TCR03 equals FFFFH). Interrupt signal (INTTM02) to be generated at TCR02 = 0000H.
Operation stop	The duty of the mask waveform is set to 0%. TDR02 = 0000H TDR03 is setting not required. **Caution Setting must finish before the TCR02 value reaches 0000H.** Wait for an interrupt signal (INTTM02) The TTmn bit (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.	The P32/TO03 pin outputs the result of ANDing the TO03 and TO07 outputs (a remote control output (carrier waveform) until TCR03 reaches 0000H; a low-level remote control output after TCR03 reaches 0000H). Interrupt signal (INTTM02) to be generated at TCR02 = 0000H. TEmn, TEmp, TEmq = 0, and count operation stops.
	The TOEmn and TOEmp bits are cleared to 0 and TOmn and TOmp bits are cleared to 0 Note	The TOmp pin is clear to low-level.
Transmission restart	To resume transmission, set the TOEmp bit of timer output enable register m (TOEm) to 1. (The TOEmn bit remains 0.)	

Operation is resumed.

Note If these bits are not used by any TAU channel, clock supply may be stopped by clearing the TAU0EN bit of peripheral enable register 0 (PER0) to 0. In this case, to resume transmission, the settings for transmission must be re-specified after the power is turned on.

Remark m: Unit number (m = 0), n: Master channel number (n = 2, 6)
p: Slave channel number (p = 3, 7)
(When mask waveform: n = 2, n = 3; When carrier waveform: n = 6, p = 7)

CHAPTER 7 REAL-TIME CLOCK

7.1 Functions of Real-time Clock

The real-time clock has the following features.

- Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
- Constant-period interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm interrupt function (alarm: week, hour, minute)
- Pin output function of 1 Hz

The real-time clock interrupt signal (INTRTC) can be utilized for wakeup from STOP mode and triggering an A/D converter's SNOOZE mode.

Caution The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock. When the low-speed on-chip oscillator clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available.
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}.

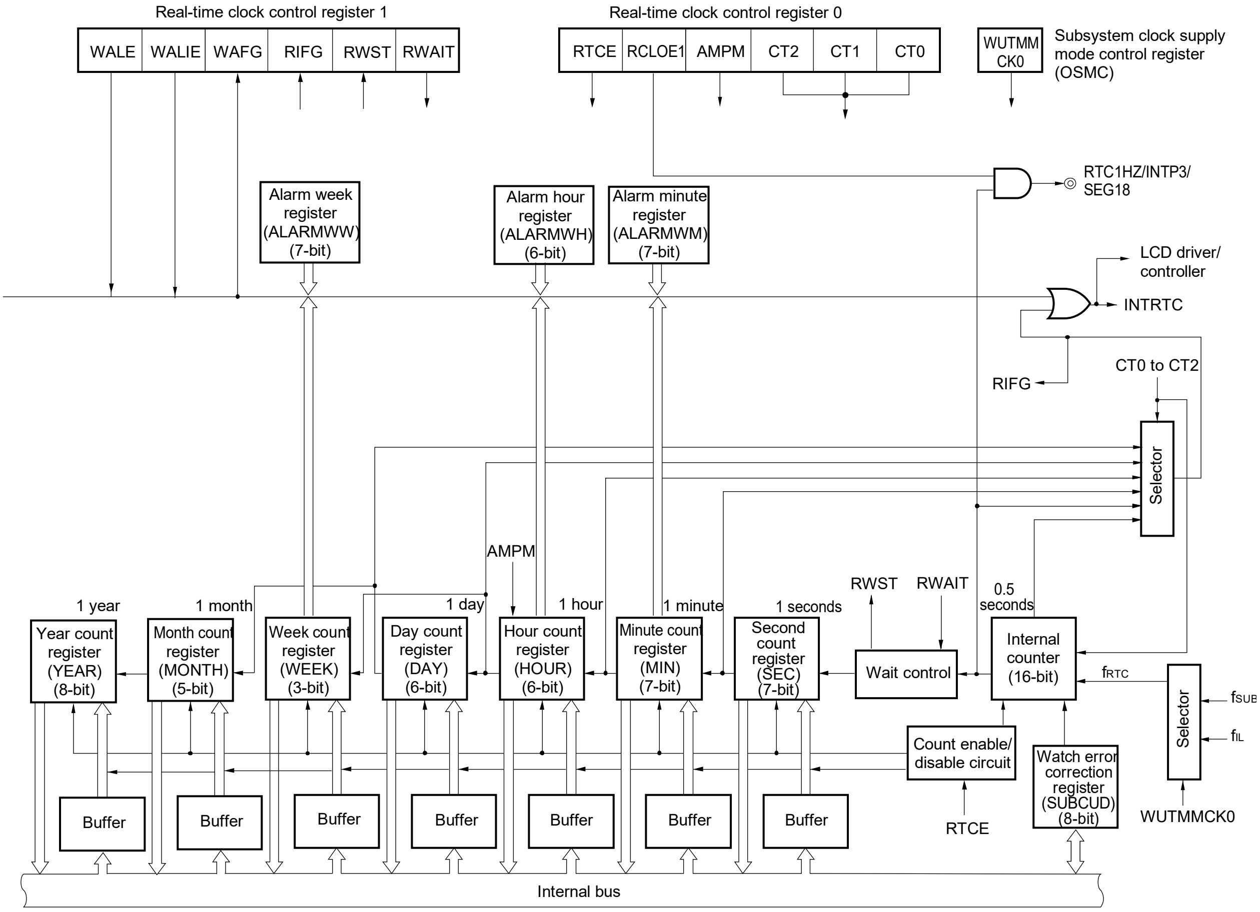
7.2 Configuration of Real-time Clock

The real-time clock includes the following hardware.

Table 7-1. Configuration of Real-time Clock

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 7-1. Block Diagram of Real-time Clock



Caution **The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock. When the low-speed on-chip oscillator clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available**

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}.

7.3 Registers Controlling Real-time Clock

The real-time clock is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- Real-time clock control register 0 (RTCC0)
- Real-time clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Watch error correction register (SUBCUD)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm week register (ALARMWW)
- Port mode register 3 (PM3)
- Port register 3 (P3)

7.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the real-time clock is used, be sure to set bit 7 (RTCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 7-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

RTCEN	Real-time clock (RTC) and 12-bit interval timer	LCD driver/controller and clock output/buzzer output controller	
		When subsystem clock (f_{SUB}) is selected	When subsystem clock (f_{SUB}) is not selected
0	Stops input clock supply. ● SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. ● The real-time clock (RTC) and 12-bit interval timer are in the reset status.	Stops input clock and subsystem clock supply. ● SFR used by the LCD driver/controller and clock output/buzzer output can be read and written.	Enables input clock and main system clock supply. ● SFR used by the LCD driver/controller and clock output/buzzer output can be read and written.
1	Enables input clock supply. ● SFR used by the real-time clock (RTC) and 12-bit interval timer can be read and written.	Enables input clock and subsystem clock supply. ● SFR used by the LCD driver/controller and clock output/buzzer output can be read and written.	

Cautions 1. When using the real-time clock, first set the RTCEN bit to 1 and then set the following registers, while oscillation of the count clock (f_{RTC}) is stable. If RTCEN = 0, writing to the control registers of the real-time clock is ignored, and, even if the registers are read, only the default values are read (except for the subsystem clock supply mode control register (OSMC), port mode register 3 (PM3), port register 3 (P3)).

- **Real-time clock control register 0 (RTCC0)**
- **Real-time clock control register 1 (RTCC1)**
- **Second count register (SEC)**
- **Minute count register (MIN)**
- **Hour count register (HOUR)**
- **Day count register (DAY)**
- **Week count register (WEEK)**
- **Month count register (MONTH)**
- **Year count register (YEAR)**
- **Watch error correction register (SUBCUD)**
- **Alarm minute register (ALARMWM)**
- **Alarm hour register (ALARMWH)**
- **Alarm week register (ALARMWW)**

2. The subsystem clock supply to peripheral functions other than the real-time clock, 12-bit interval timer, and LCD driver/controller can be stopped in STOP

mode and HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the subsystem clock supply mode control register (OSMC) to 1. In this case, set the RTCEN bit of the PER0 register to 1 and the other bits (bits 0 to 6) to 0.

3. Be sure to clear the bits 1, 3, 4 and 6 to 0.

7.3.2 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the real-time clock operation clock (f_{RTC}).

In addition, by stopping clock functions that are an 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 7-3. 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 Note	Selection of operation clock for real-time clock, 12-bit interval timer, and LCD river/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.

Cautions 1. The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock. When the low-speed on-chip oscillator clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available. However, the constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SUB}/f_{IL}.

2. The subsystem clock and low-speed on-chip oscillator clock can only be switched by using the WUTMMCK0 bit if the real-time clock, 12-bit interval timer, and LCD driver/controller are all stopped.
These are stopped as follows:

Real-time clock: 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 interval timer control register (ITMC)
SCOC:	Bit 6 of LCD mode register 1 (LCDM1)
VLCON:	Bit 5 of LCD mode register 1 (LCDM1)

7.3.3 Real-time clock control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time clock operation, control the RTC1HZ pin, and set a 12- or 24-hour system and the constant-period interrupt function.

The RTCC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-4. Format of Real-time Clock Control Register 0 (RTCC0)

Address: FFF9DH After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1	0	AMPM	CT2	CT1	CT0

RTCE	Real-time clock operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1	RTC1HZ pin output control
0	Disables output of the RTC1HZ pin (1 Hz).
1	Enables output of the RTC1HZ pin (1 Hz).

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
● Rewrite the AMPM bit value after setting the RWAIT bit (bit 0 of real-time clock control register 1 (RTCC1)) to 1. If the AMPM bit value is changed, the values of the hour count register (HOUR) change according to the specified time system. ● Table 7-2 shows the displayed time digits that are displayed.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period 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.			

Cautions 1. Do not change the value of the RCLOE1 bit when RTCE = 1.

2. 1 Hz is not output even if RCLOE1 is set to 1 when RTCE = 0.

Remark ×: don't care

7.3.4 Real-time clock control register 1 (RTCC1)

The RTCC1 register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.

The RTCC1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-5. Format of Real-time Clock Control Register 1 (RTCC1) (1/2)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
When setting a value to the WALE bit while the counter operates (RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting each alarm register (WALIE flag of real-time clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm week register (ALARMWW)), set match operation to be invalid (“0”) for the WALE bit.	

WALIE	Control of alarm interrupt (INTRTC) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to “1” one 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.	

Figure 7-5. Format of Real-time Clock Control Register 1 (RTCC1) (2/2)

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.	

RWST	Wait status flag of real-time clock
0	Counter is operating.
1	Mode to read or write counter value
This status flag indicates whether the setting of the RWAIT bit is valid. Before reading or writing the counter value, confirm that the value of this flag is 1.	

RWAIT	Wait control of real-time clock
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value
This bit controls the operation of the counter. Be sure to write “1” to it to read or write the counter value. As the counter (16-bit) is continuing to run, complete reading or writing within one second and turn back to 0. <R> 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). Notes 1, 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.	

Notes 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”.

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 may take two clock time of the operation clock (f_{RTC}), until RWST bit is set to “1”.

Caution If writing is performed to the RTCC1 register with a 1-bit manipulation instruction, the RIFG flag and WAFG flag may be cleared. Therefore, to perform writing to the RTCC1 register, be sure to use an 8-bit manipulation instruction. To prevent the RIFG flag and WAFG flag from being cleared during writing, disable writing by setting 1 to the corresponding bit. If the RIFG flag and WAFG flag are not used and the value may be changed, the RTCC1 register may be written by using a 1-bit manipulation instruction.

Remark 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.

7.3.5 Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the counter (16-bit) overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Set a decimal value of 00 to 59 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-6. Format of Second Count Register (SEC)

Address: FFF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Cautions 1. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

2. The internal counter (16 bits) is cleared when the second count register (SEC) is written.

7.3.6 Minute count register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 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 clears this register to 00H.

Figure 7-7. Format of Minute Count Register (MIN)

Address: FFF93H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

7.3.7 Hour count register (HOUR)

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using bit 3 (AMPM) of real-time clock control register 0 (RTCC0).

If the AMPM bit value is changed, the values of the HOUR register change according to the specified time system.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Figure 7-8. Format of Hour Count Register (HOUR)

Address: FFF94H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Cautions 1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

Table 7-2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 7-2. Displayed Time Digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 1)	
Time	HOUR Register	Time	HOUR Register
0	00H	12 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	12 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

The HOUR register value is set to 12-hour display when the AMPM bit is "0" and to 24-hour display when the AMPM bit is "1".

In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

7.3.8 Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.

It counts up when the hour counter overflows.

This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Even if the hour count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 31 to this register in BCD code.

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

Reset signal generation clears this register to 01H.

Figure 7-9. Format of Day Count Register (DAY)

Address: FFF96H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

7.3.9 Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays. It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Set a decimal value of 00 to 06 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-10. Format of Week Count Register (WEEK)

Address: FFF95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Cautions 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

7.3.10 Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code.

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

Reset signal generation clears this register to 01H.

Figure 7-11. Format of Month Count Register (MONTH)

Address: FFF97H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

7.3.11 Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month count register (MONTH) overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Even if the MONTH register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-12. Format of Year Count Register (YEAR)

Address: FFF98H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in the section 7.4.3 Reading/writing real-time clock.

7.3.12 Watch error correction register (SUBCUD)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value that overflows from the internal counter (16-bit) to the second count register (SEC) (reference value: 7FFFH).

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

Reset signal generation clears this register to 00H.

Figure 7-13. Format of Watch Error Correction Register (SUBCUD)

Address: FFF99H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch error only when the second digits are at 00 (every 60 seconds).

Writing to the SUBCUD register at the following timing is prohibited.
- When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H
- When DEV = 1 is set: For a period of SEC = 00H

F6	Setting of watch error correction value
0	Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.

When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected. * is 0 or 1.
/F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100).

Range of correction value: (when F6 = 0) 2, 4, 6, 8, … , 120, 122, 124
(when F6 = 1) –2, –4, –6, –8, … , –120, –122, –124

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	–189.2 ppm to 189.2 ppm	–63.1 ppm to 63.1 ppm
Maximum excludes quantization error	± 1.53 ppm	± 0.51 ppm
Minimum resolution	± 3.05 ppm	± 1.02 ppm

Remark If a correctable range is -63.1 ppm or lower and 63.1 ppm or higher, set 0 to DEV.

7.3.13 Alarm minute register (ALARMWM)

This register is used to set minutes of alarm.

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

Reset signal generation clears this register to 00H.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 7-14. Format of Alarm Minute Register (ALARMWM)

Address: FFF9AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

7.3.14 Alarm hour register (ALARMWH)

This register is used to set hours of alarm.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 7-15. Format of Alarm Hour Register (ALARMWH)

Address: FFF9BH After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

7.3.15 Alarm week register (ALARMWW)

This register is used to set date of alarm.

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

Reset signal generation clears this register to 00H.

Figure 7-16. Format of Alarm Week Register (ALARMWW)

Address: FFF9CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

7.3.16 Port mode register 3 (PM3)

The PM3 register can be set by a 1-bit or 8-bit manipulation instruction.

Reset signal generation sets this register to FFH.

When using the port 3 as the RTC1HZ pin for output of 1 Hz, set the PM30 bit to 0.

Figure 7-17. Format of Port Mode Register 3 (PM3)

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

7.3.17 Port register 3 (P3)

The PM3 register can be set by a 1-bit or 8-bit manipulation instruction.

Reset signal generation sets this register to 00H.

When using the port 3 as 1-Hz output to the RTC1Hz pin, set the P30 bit to 0.

Figure 7-18. Format of Port Register 3 (P3)

Address: FFF03H After reset: 00H R/W

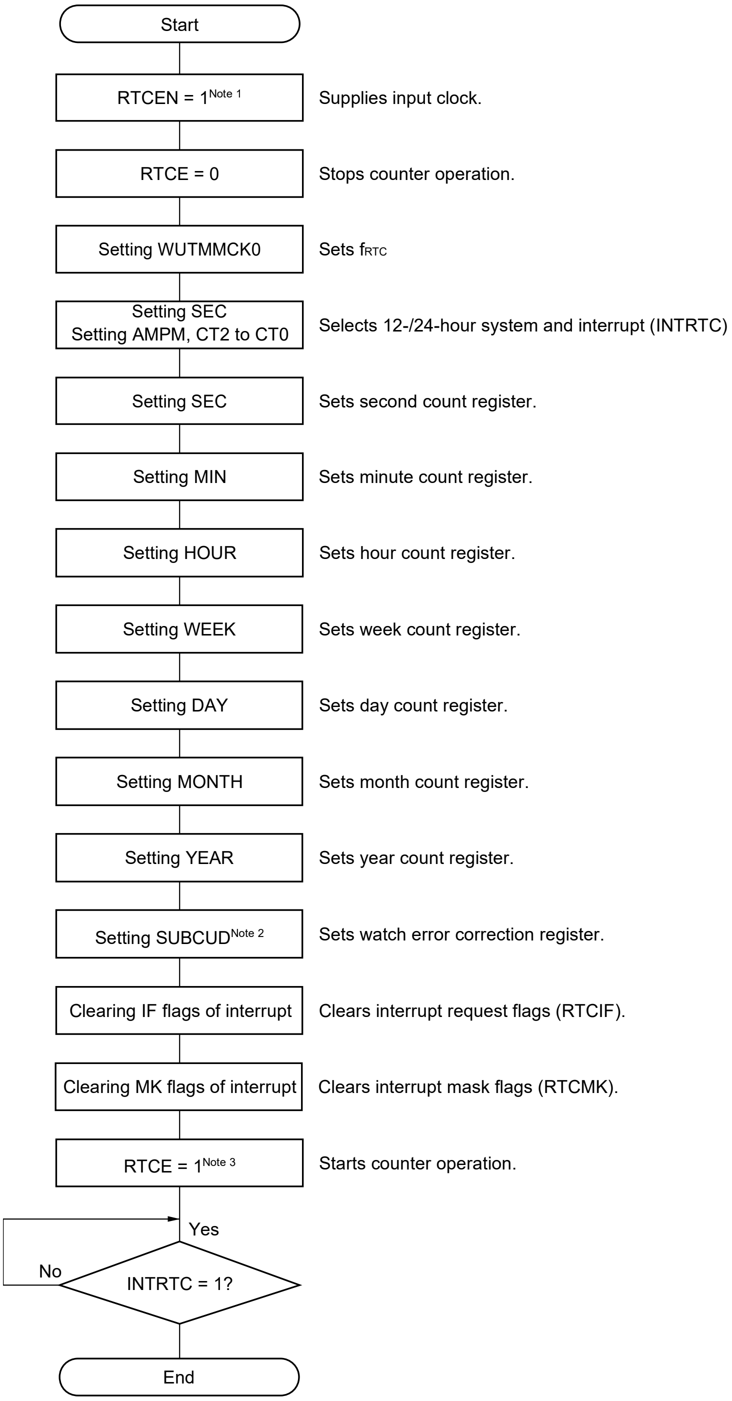
Symbol	7	6	5	4	3	2	1	0
P3	P37	P36	P35	P34	P33	P32	P31	P30

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

7.4 Real-time Clock Operation

7.4.1 Starting operation of real-time clock

Figure 7-19. Procedure for Starting Operation of Real-time Clock



Notes 1. First set the RTCEN bit to 1, while oscillation of the count clock (f_{RTC}) is stable.

2. Set up the SUBCUD register only if the watch error must be corrected. For details about how to calculate the correction value, see **7.4.6 Example of watch error correction of real-time clock**.

3. Confirm the procedure described in **7.4.2 Shifting to HALT/STOP mode after starting operation** when shifting to HALT/STOP mode without waiting for INTRTC = 1 after RTCE = 1.

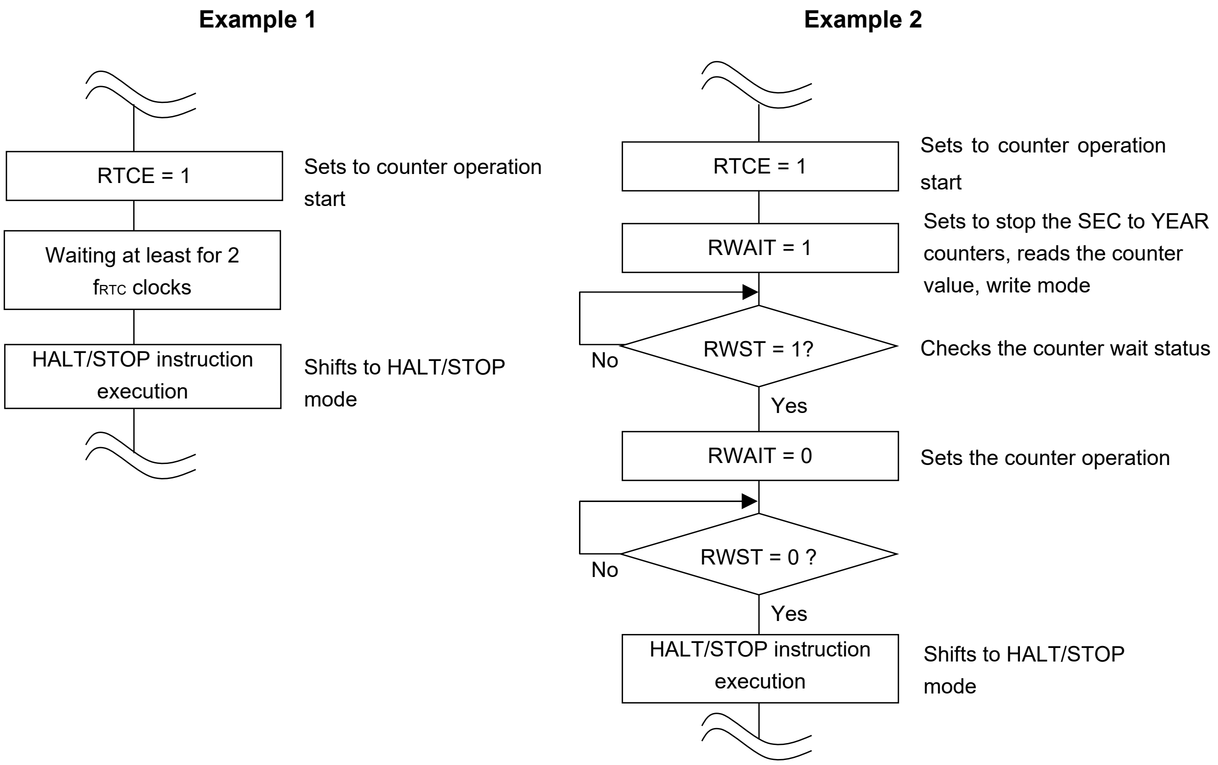
7.4.2 Shifting to HALT/STOP mode after starting operation

Perform one of the following processing when shifting to HALT/STOP mode immediately after setting the RTCE bit to 1.

However, after setting the RTCE bit to 1, this processing is not required when shifting to HALT/STOP mode after the first INTRTC interrupt has occurred.

- Shifting to HALT/STOP mode when at least two input clocks (f_{RTC}) have elapsed after setting the RTCE bit to 1 (see **Figure 7-20**, **Example 1**).
- Checking by polling the RWST bit to become 1, after setting the RTCE bit to 1 and then setting the RWAIT bit to 1. Afterward, setting the RWAIT bit to 0 and shifting to HALT/STOP mode after checking again by polling that the RWST bit has become 0 (see **Figure 7-20**, **Example 2**).

Figure 7-20. Procedure for Shifting to HALT/STOP Mode After Setting RTCE bit to 1


Example 1

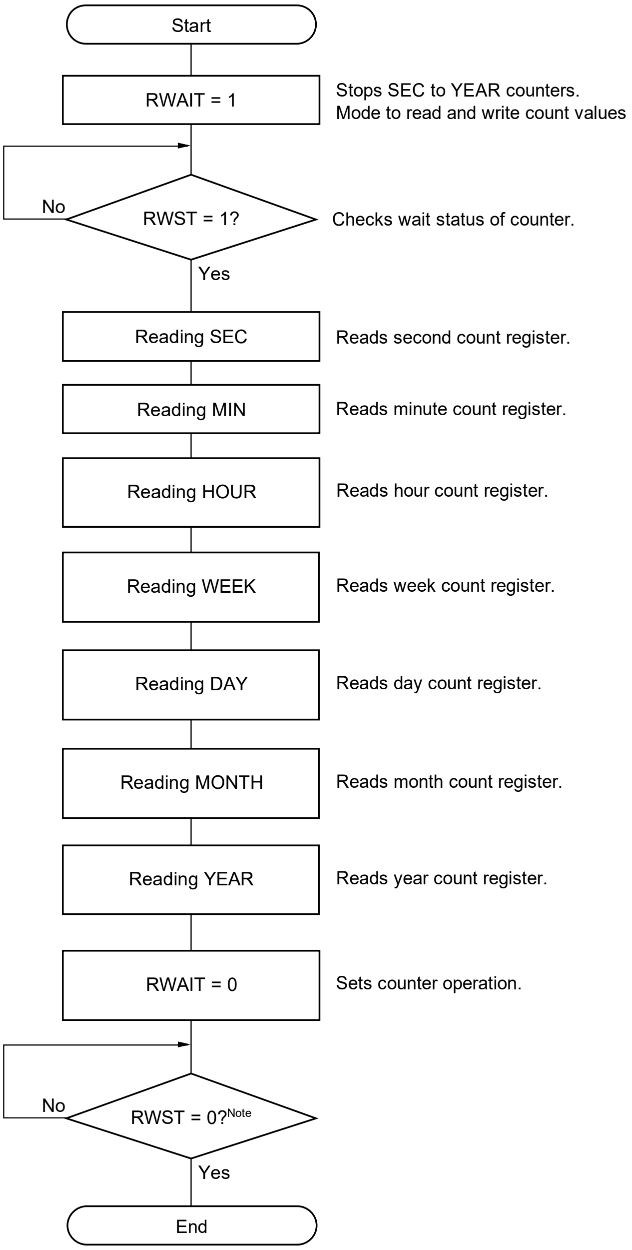
Example 2

7.4.3 Reading/writing real-time clock

Read or write the counter after setting 1 to RWAIT first.

Set RWAIT to 0 after completion of reading or writing the counter.

Figure 7-21. Procedure for Reading Real-time Clock



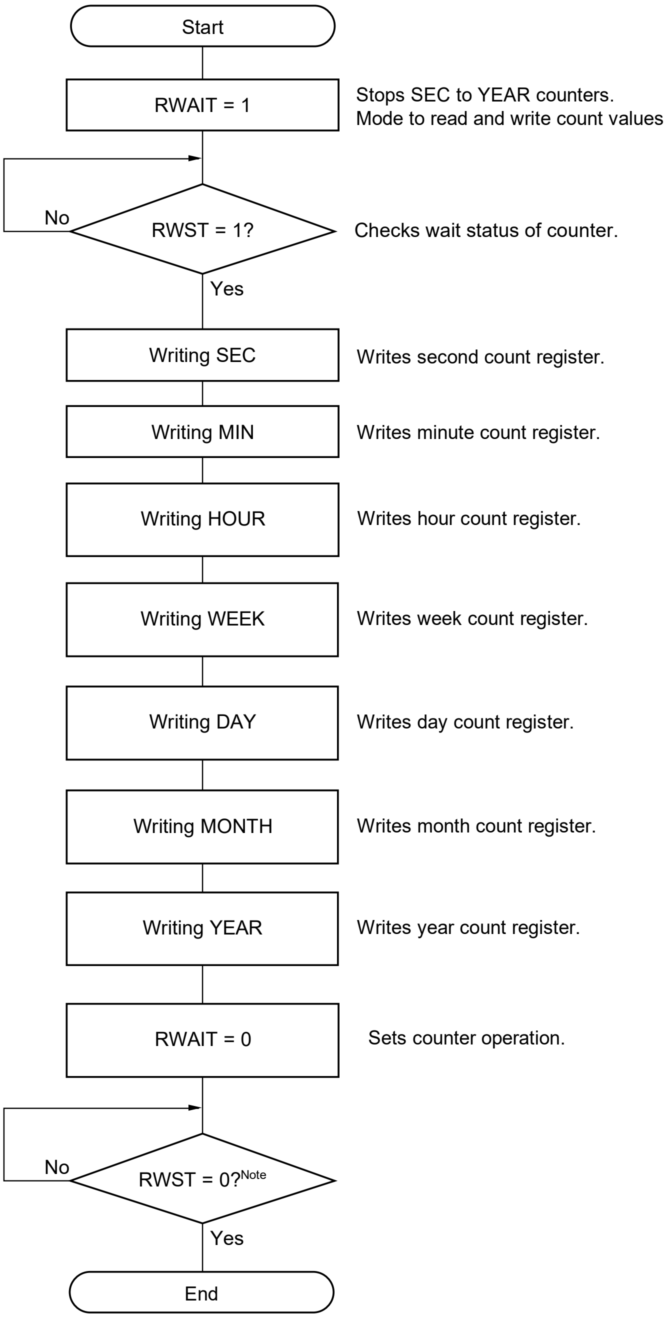
Note Be sure to confirm that RWST = 0 before setting STOP mode.

<R> **Caution Complete the series of process of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second. When reading to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.**

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence.

All the registers do not have to read and only some registers may be read.

Figure 7-22. Procedure for Writing Real-time Clock



Note Be sure to confirm that RWST = 0 before setting STOP mode.

Cautions 1. Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second. When writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt. <R>

2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counter operates (RTCE = 1), rewrite the values of the MIN register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the MIN register.

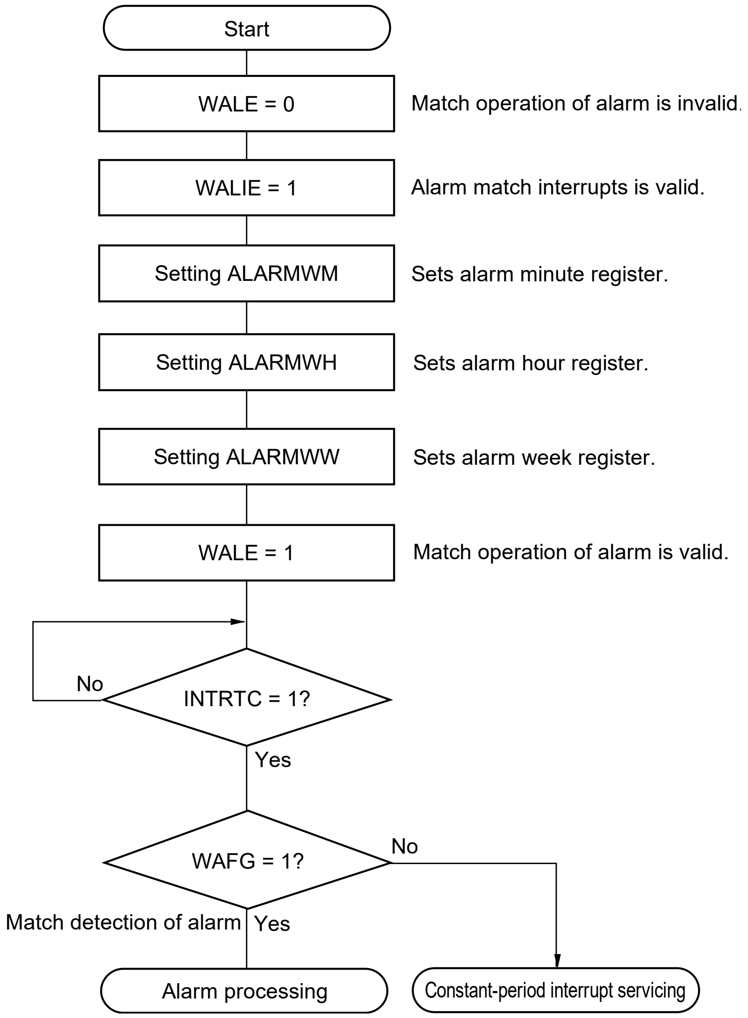
Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be written in any sequence.

All the registers do not have to be set and only some registers may be written.

7.4.4 Setting alarm of real-time clock

Set time of alarm after setting 0 to WALE (alarm operation invalid.) first.

Figure 7-23. Alarm processing Procedure

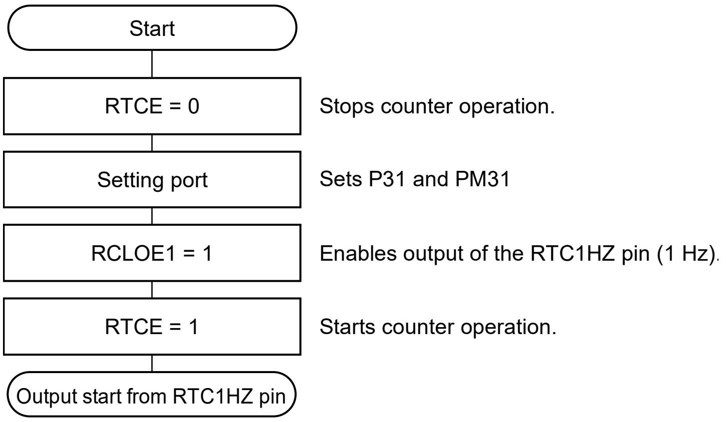


Remarks 1. The alarm week register (ALARMWW), alarm hour register (ALARMWH), and alarm week register (ALARMWW) may be written in any sequence.

2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

7.4.5 1 Hz output of real-time clock

Figure 7-24. 1 Hz Output Setting Procedure



Cautions First set the RTCEN bit to 1, while oscillation of the input clock (f_{SUB}) is stable.

7.4.6 Example of watch error correction of real-time clock

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the watch error correction register.

Example of calculating the correction value

The correction value used when correcting the count value of the internal counter (16-bit) is calculated by using the following expression.

Set the DEV bit to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(When DEV = 0)

Correction value[Note] = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60 ÷ 3

(When DEV = 1)

Correction value[Note] = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60

Note The correction value is the watch error correction value calculated by using bits 6 to 0 of the watch error correction register (SUBCUD).

(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) – 1} × 2
(When F6 = 1) Correction value = – {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), watch error correction is not performed. “*” is 0 or 1.
/F5 to /F0 are bit-inverted values (000011 when 111100).

Remarks 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

2. The oscillation frequency is the input clock (f_{RTC}).
It can be calculated from the output frequency of the RTC1HZ pin × 32768 when the watch error correction register is set to its initial value (00H).

3. The target frequency is the frequency resulting after correction performed by using the watch error correction register.

Correction example 1

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32.768 kHz from the PCLBUZ0 pin, or by outputting about 1 Hz from the RTC1HZ pin when the watch error correction register (SUBCUD) is set to its initial value (00H).

Note See **7.4.5 1 Hz output of real-time clock** for the setting procedure of the RTC1Hz output, and see **9.4 Operations of Clock Output/Buzzer Output Controller** for the setting procedure of outputting about 32 kHz from the PCLBUZ0 pin.

[Calculating the correction value]

(When the output frequency from the PCLBUZ0 pin is 32772.3 Hz)

Assume the target frequency to be 32768 Hz (32772.3 Hz–131.2 ppm) and DEV to be 0, because the correctable range of –131.2 ppm is –63.1 ppm or lower.

The expression for calculating the correction value when DEV is 0 is applied.

Correction value = Number of correction counts in 1 minute ÷ 3
= (Oscillation frequency ÷ target frequency – 1) × 32768 × 60 ÷ 3
= (32772.3 ÷ 32768 – 1) × 32768 × 60 ÷ 3
= 86

[Calculating the values to be set to (F6 to F0)]

(When the correction value is 86)

If the correction value is 0 or larger (when slowing), assume F6 to be 0.

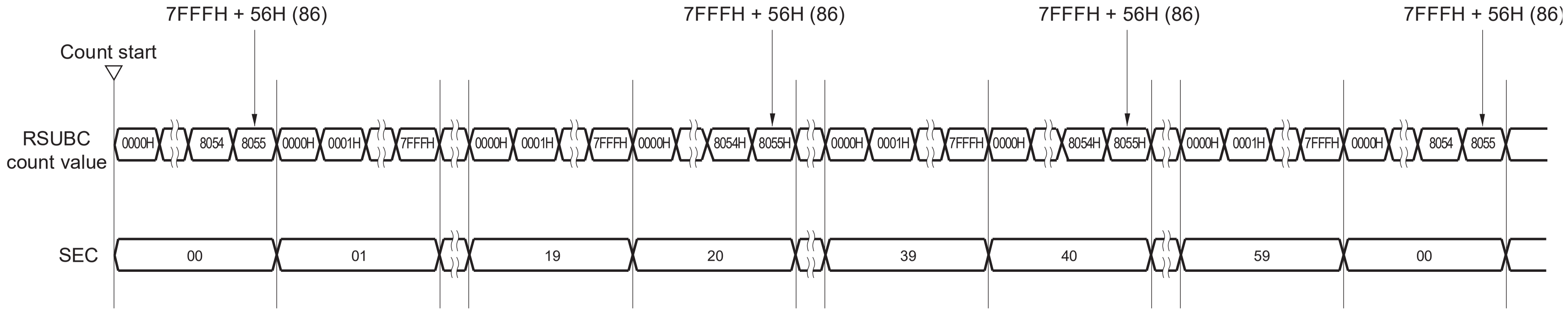
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

{ (F5, F4, F3, F2, F1, F0) – 1} × 2 = 86
(F5, F4, F3, F2, F1, F0) = 44
(F5, F4, F3, F2, F1, F0) = (1, 0, 1, 1, 0, 0)

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm), setting the correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of the SUBCUD register: 0101100) results in 32768 Hz (0 ppm).

Figure 7-25 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 7-25. Correction Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)



Correction example 2

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 1 Hz from the RTC1HZ pin when the watch error correction register (SUBCUD) is set to its initial value (00H).

Note See **7.4.5 1 Hz output of real-time clock** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 0.9999817 Hz)

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

$$\begin{aligned}\text{Correction value} &= \text{Number of correction counts in 1 minute}\\ &= (\text{Oscillation frequency} \div \text{Target frequency} - 1) \times 32768 \times 60\\ &= (32767.4 \div 32768 - 1) \times 32768 \times 60\\ &= -36\end{aligned}$$

[Calculating the values to be set to (F6 to F0)]

(When the correction value is –36)

If the correction value is 0 or less (when quickening), assume F6 to be 1.

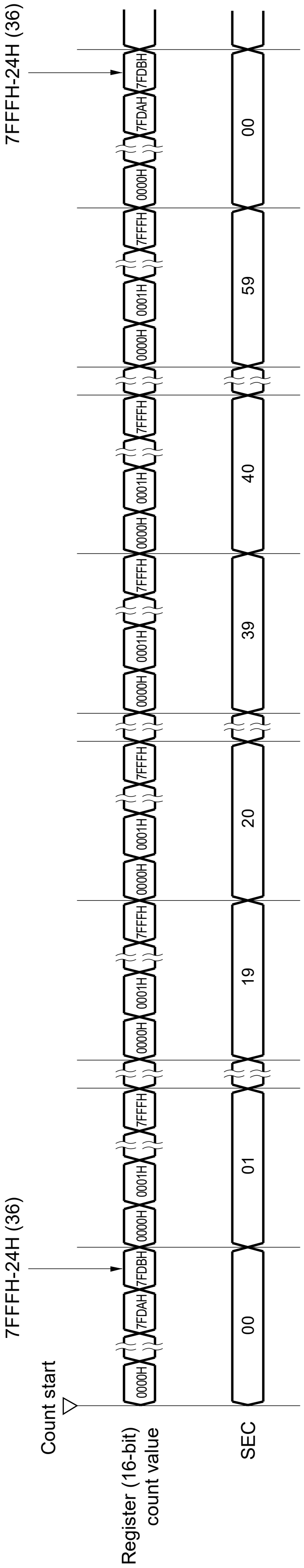
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

$$\begin{aligned} -\{(/F5, /F4, /F3, /F2, /F1, /F0) - 1\} \times 2 &= -36\\ (/F5, /F4, /F3, /F2, /F1, /F0) &= 17\\ (/F5, /F4, /F3, /F2, /F1, /F0) &= (0, 1, 0, 0, 0, 1)\\ (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 1, 0)\end{aligned}$$

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is –36 (bits 6 to 0 of the SUBCUD register: 1101110) results in 32768 Hz (0 ppm).

Figure 7-26 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

Figure 7-26. Correcting Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)



CHAPTER 8 12-BIT INTERVAL TIMER

8.1 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.

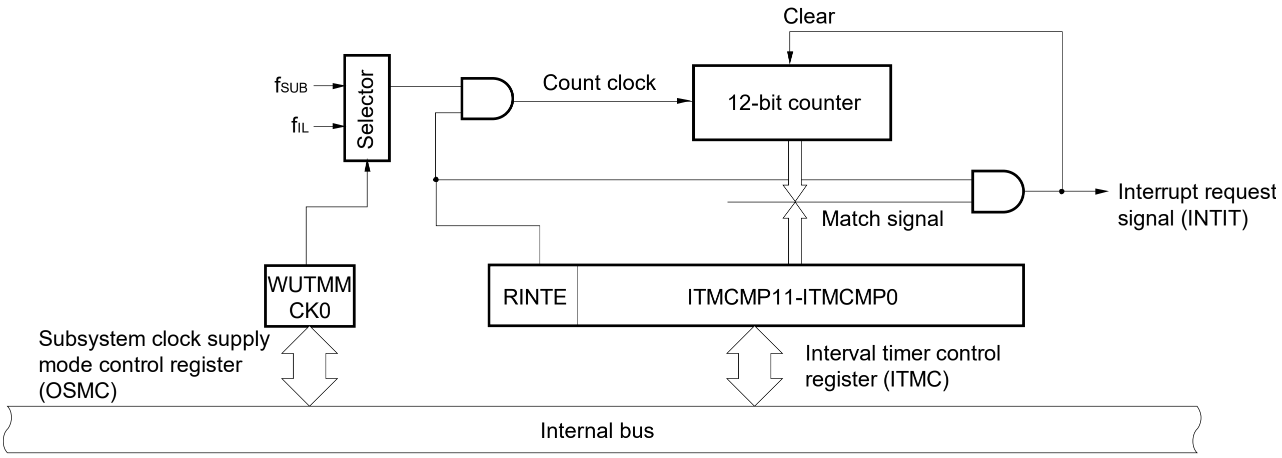
8.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 8-1. Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 0 (PER0)
	Subsystem clock supply mode control register (OSMC)
	Interval timer control register (ITMC)

Figure 8-1. Block Diagram of 12-bit Interval Timer



8.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- Interval timer control register (ITMC)

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 12-bit interval timer is used, be sure to set bit 7 (RTCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 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	RTCEN	0	ADCEN	0	0	SAU0EN	0	TAU0EN

RTCEN	Control of clock supply to real-time clock (RTC) and 12-bit interval timer	LCD driver/controller and clock output/buzzer output controller	
		When subsystem clock (f_{SUB}) is selected	When subsystem clock (f_{SUB}) is not selected
0	Stops input clock supply. ● SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. ● The real-time clock (RTC) and 12-bit interval timer are in the reset status.	Stops input clock and subsystem clock supply. ● SFR used by the LCD driver/controller and clock output/buzzer output can be read and written.	Enables input clock and main system clock supply. ● SFR used by the LCD driver/controller and clock output/buzzer output can be read and written.
1	Enables input clock supply. ● SFR used by the real-time clock (RTC) and 12-bit interval timer can be read and written.	Enables input clock and subsystem clock supply. ● SFR used by the LCD driver/controller and clock output/buzzer output can be read and written.	

Cautions 1. When using the 12-bit interval timer, first set the RTCEN bit to 1, while oscillation of the input clock (f_{RTC}) is stable. If RTCEN = 0, writing to a control register of the real-time clock, 12-bit interval timer, or LCD driver/controller is ignored, and, even if the register is read, only the default value is read (except the subsystem clock supply mode control register (OSMC)).

2. Clock supply to peripheral functions other than the real-time clock, 12-bit interval timer, and LCD driver/controller can be stopped in STOP mode and HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the subsystem clock supply mode control register (OSMC) to 1. In this case, set the RTCEN bit of the PER0 register to 1 and the other bits (bits 0 to 6) to 0.

3. Be sure to clear the bits 1, 3, 4 and 6 to 0.

8.3.2 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 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

WUTMMCK0 Note	Selection of operation clock for real-time clock, 12-bit interval timer, and LCD river/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, 12-bit interval timer, and LCD driver/controller are all stopped.
These are stopped as follows:

Real-time clock: Set the RTCE bit to 0.
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 interval timer control register (ITMC)
SCOC: Bit 6 of LCD mode register 1 (LCDM1)
VLCON: Bit 5 of LCD mode register 1 (LCDM1)

8.3.3 Interval timer control register (ITMC)

This register is used to set up the starting and stopping of the 12-bit interval timer operation and to specify the timer compare value.

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

Reset signal generation clears this register to 0FFFH.

Figure 8-4. Format of Interval Timer Control Register (ITMC)

Address: FFF90H After reset: 0FFFH R/W

Symbol	15	14	13	12	11 to 0
ITMC	RINTE	0	0	0	ITCMP11 to ITCMP0

RINTE	12-bit interval timer operation control
0	Count operation stopped (count clear)
1	Count operation started

ITCMP11 to ITCMP0	Specification of the 12-bit interval timer compare value
001H	These bits generate an interrupt at the fixed cycle (count clock cycles x (ITCMP setting + 1)).
• • •	
FFFH	
000H	Setting prohibit

Example interrupt cycles when 001H or FFFH is specified for ITCMP11 to ITCMP0

- ITCMP11 to ITCMP0 = 001H, count clock: when f_{SUB} = 32.768 kHz
  1/32.768 [kHz] × (1 + 1) = 0.06103515625 [ms] ≈ 61.03 [μs]
- ITCMP11 to ITCMP0 = FFFH, count clock: when f_{SUB} = 32.768 kHz
  1/32.768 [kHz] × (4095 + 1) = 125 [ms]

Cautions 1. Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. When the operation starts (from 0 to 1) again, clear the ITIF flag, and then enable the interrupt servicing.

2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit.

3. When setting the RINTE bit to start operation of the counter after returning from a standby mode and then shifting to a standby mode again, either confirm that the value written to the RINTE bit has been applied, or make sure that at least one count clock cycle elapses between returning from a standby mode and shifting to a standby mode again.

4. Only change the setting of the ITCMP11 to ITCMP0 bits when RINTE = 0.
However, it is possible to change the settings of the ITCMP11 to ITCMP0 bits at the same time as when changing RINTE from 0 to 1 or 1 to 0.

8.4 12-bit Interval Timer Operation

8.4.1 12-bit interval timer operation timing

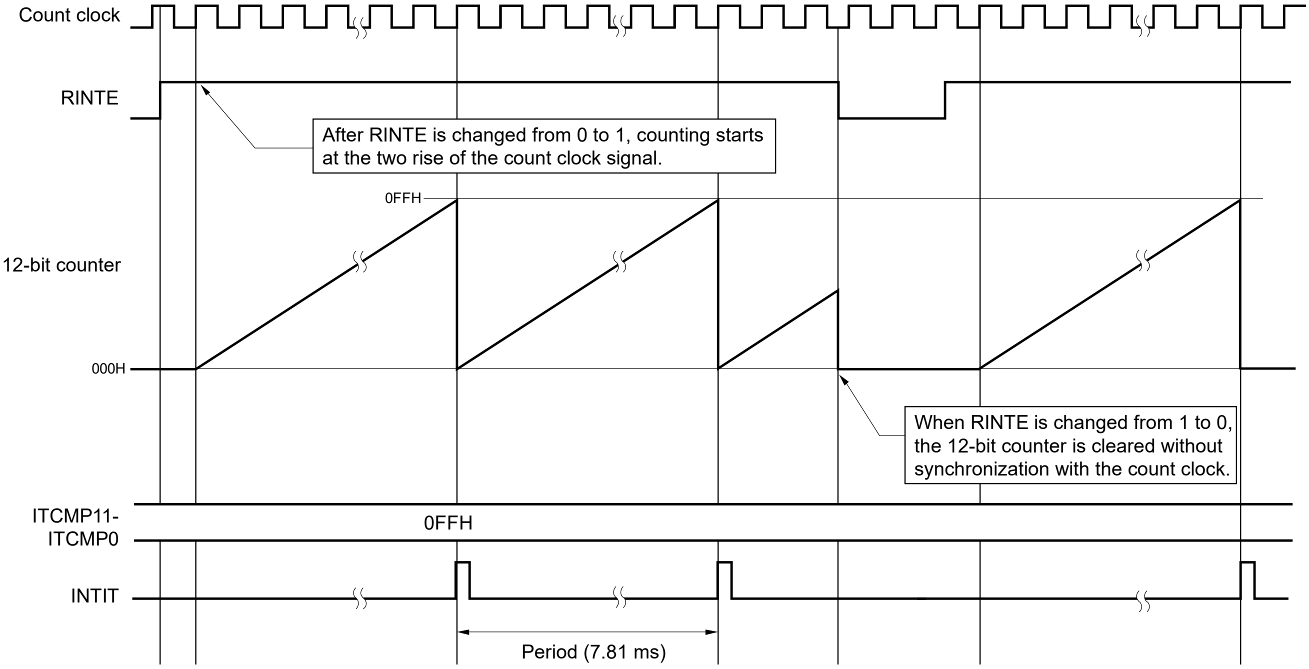
The count value specified for the ITCMP11 to ITCMP0 bits is used as an interval to operate an 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

When the RINTE bit is set to 1, the 12-bit counter starts counting.

When the 12-bit counter value matches the value specified for the ITCMP11 to ITCMP0 bits, the 12-bit counter value is cleared to 0, counting continues, and an interrupt request signal (INTIT) is generated at the same time.

The basic operation of the 12-bit interval timer is as follows.

Figure 8-5. 12-bit Interval Timer Operation Timing (ITCMP11 to ITCMP0 = 0FFH, count clock: f_{SUB} = 32.768 kHz)

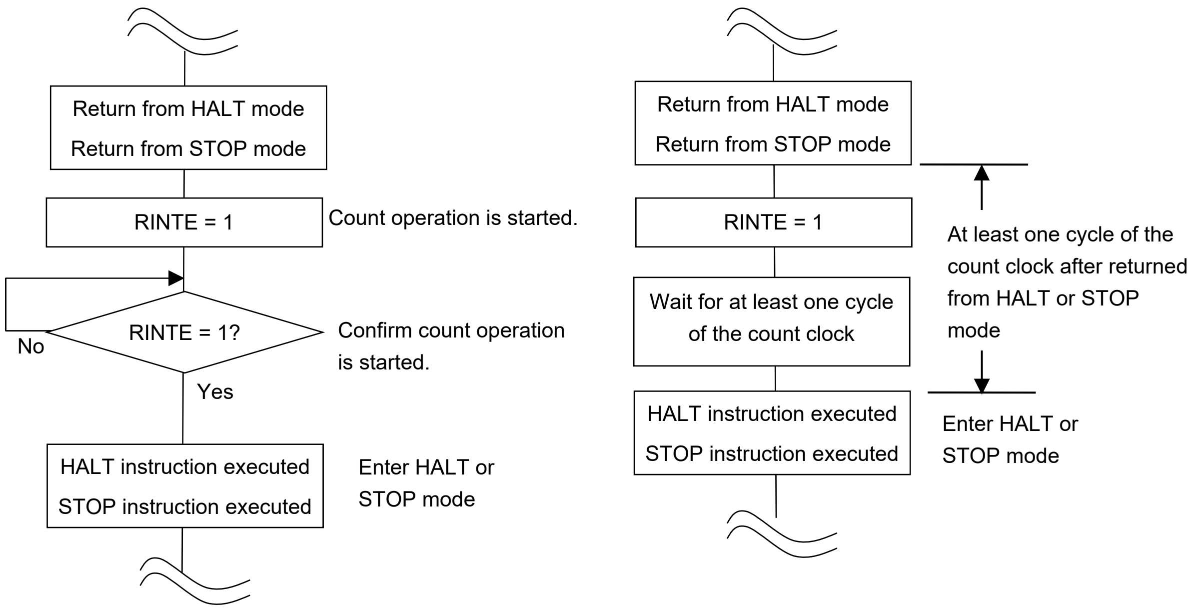


8.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

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 8-6).
- 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 8-6).

Figure 8-6. Procedure of Entering to HALT or STOP Mode after Setting RINTE to 1



Example 1 Example 2

CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

9.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral ICs.

Buzzer output is a function to output a square wave of buzzer frequency.

One pin can be used to output a clock or buzzer sound.

Two output pins, PCLBUZ0 and PCLBUZ1, are available.

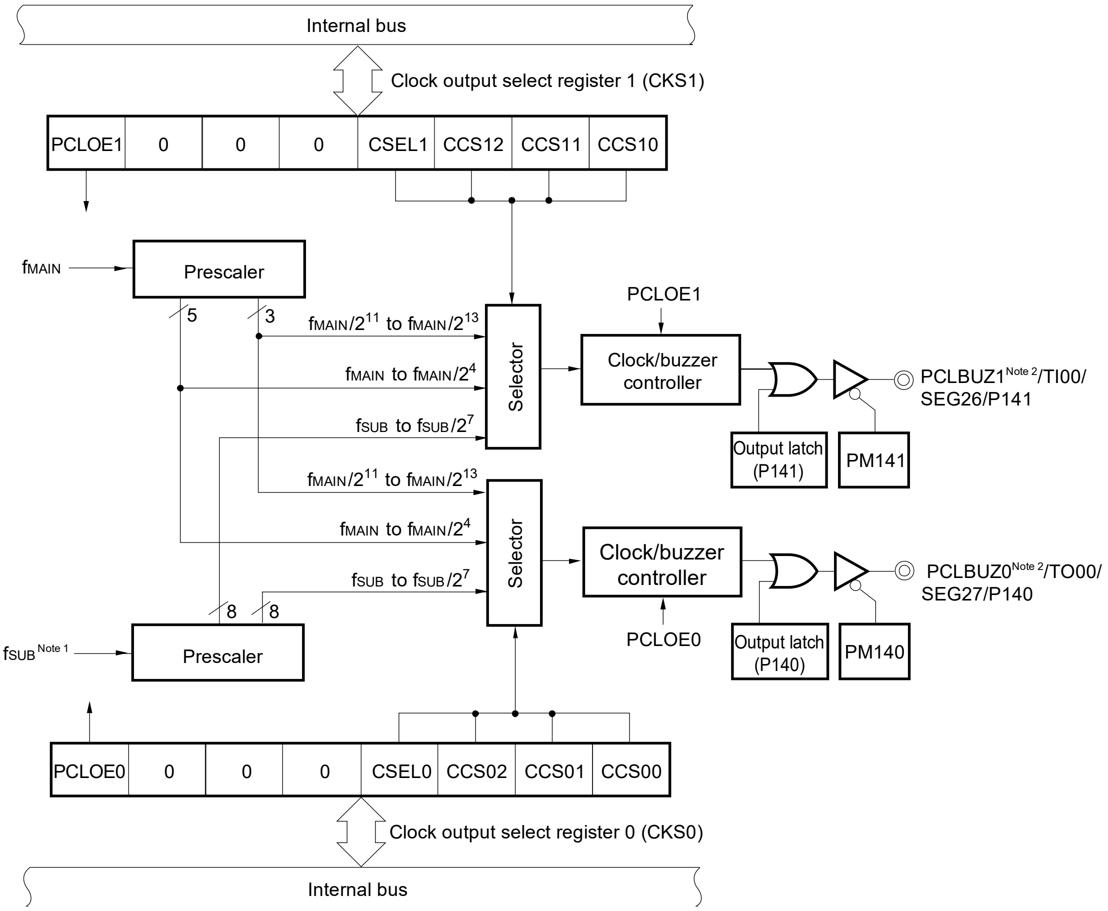
The PCLBUZn pin outputs a clock selected by clock output select register n (CKSn).

Figure 9-1 shows the block diagram of clock output/buzzer output controller.

Caution It is not possible to output the subsystem clock (f_{SUB}) from the PCLBUZn pin while the RTCLPC bit of the subsystem clock supply mode control register (OSMC), which controls the supply of the subsystem clock, is set to 1 and moreover while HALT mode is set with the subsystem clock (f_{SUB}) selected as CPU clock.

Remark n = 0, 1

Figure 9-1. Block Diagram of Clock Output/Buzzer Output Controller



Notes 1. Do not select fSUB as the clock output from the clock output/buzzer output controller when the WUTMMCK0 bit of the OSMC register is set to 1.

2. For output frequencies available from PCLBUZ0 and PCLBUZ1, refer **27.4 AC Characteristics**.

Remark PCLBUZ0 pin in above diagram shows the information with PIOR1 = 0.
In other cases, the name of pins, output latches (Pxx) and PMxx should be read differently (xx = 50).

9.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 9-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Peripheral enable register 0 (PER0) Clock output select registers n (CKSn) Port mode registers 5, 14 (PM5, PM14) Port registers 5, 14 (P5, P14)

9.3 Registers Controlling Clock Output/Buzzer Output Controller

The following three registers are used to control the clock output/buzzer output controller.

- Peripheral enable register 0 (PER0)
- Clock output select registers n (CKSn)
- Port mode registers 5, 14 (PM5, PM14)

9.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the clock output/buzzer output controller is used in subsystem clock (f_{SUB}), be sure to set bit 7 (RTCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 9-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

RTCEN	Real-time clock (RTC) and 12-bit interval timer	LCD driver/controller and clock output/buzzer output controller	
		When subsystem clock (f_{SUB}) is selected	When subsystem clock (f_{SUB}) is not selected
0	Stops input clock supply. ● SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. ● The real-time clock (RTC) and 12-bit interval timer are in the reset status.	Stops input clock and subsystem clock supply. ● SFR used by the LCD driver/controller and clock output/buzzer output can be read and written.	Enables input clock and main system clock supply. ● SFR used by the LCD driver/controller and clock output/buzzer output can be read and written.
1	Enables input clock supply. ● SFR used by the real-time clock (RTC) and 12-bit interval timer can be read and written.	Enables input clock and subsystem clock supply. ● SFR used by the LCD driver/controller and clock output/buzzer output can be read and written.	

Caution Be sure to clear the bits 1, 3, 4 and 6 to 0.

9.3.2 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 9-3. Format of Clock Output Select Register 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	10 MHz Note 1	Setting prohibited Note 1	Setting prohibited Note 1
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz Note 1	12 MHz 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	9.76 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			

Notes 1. Use the output clock within a range of 16 MHz. Furthermore, when using the output clock at 2.7 V ≤ V_{DD} < 4.0 V, can be use it within 8 MHz only. See **27.4 AC Characteristics** for details.

2. Do not select f_{SUB} as the clock output from the clock output/buzzer output controller when the WUTMMCK0 bit of the OSMC register is set to 1.

Cautions 1. Change the output clock after disabling clock output (PCLOEn = 0).

2. To shift to STOP mode when the main system clock is selected (CSELn = 0), set PCLOEn = 0 before executing the STOP instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn = 1 can be set because the clock can be output while the RTCLPC bit of the subsystem clock supply mode control (OSMC) register is set to 0 and moreover while STOP mode is set.

3. It is not possible to output the subsystem clock (f_{SUB}) from the PCLBUZn pin while the RTCLPC bit of the subsystem clock supply mode control register (OSMC) is set to 1 and moreover while HALT mode is set with the subsystem clock (f_{SUB}) selected as CPU clock.

Remarks 1. n = 0, 1

2. f_{MAIN}: Main system clock frequency

f_{SUB}: Subsystem clock frequency

9.3.3 Port mode registers 5, 14 (PM5, PM14)

These registers set input/output of port 5, 14 in 1-bit units.

When using the P50/INTP5/SEG7/(PCLBUZ0), P140/PCLBUZ0/TO00/SEG27 and P141/PCLBUZ1/TI00/SEG26 pins for clock output and buzzer output, clear PM50, PM140 and PM141 bits and the output latches of P50, P140 and P141 to 0.

The PM5 and PM14 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 9-4. Format of Port Mode Registers 5, 14 (PM5, PM14)

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	1	1	1	PM54	PM53	PM52	PM51	PM50

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	PM147	PM146	PM145	PM144	PM143	PM142	PM141	PM140

PMmn	Pmn pin I/O mode selection (mn = 50 to 54, 140 to 147)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

9.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.
The PCLBUZ0 pin outputs a clock/buzzer selected by the clock output select register 0 (CKS0).
The PCLBUZ1 pin outputs a clock/buzzer selected by the clock output select register 1 (CKS1).

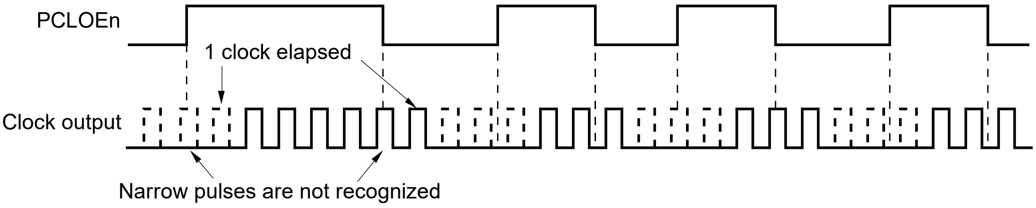
9.4.1 Operation as output pin

The PCLBUZn pin is output as the following procedure.

<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.

Remarks 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 9-5 shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.
2. n = 0, 1

Figure 9-5. Timing of Outputting Clock from PCLBUZn Pin



9.5 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 main system clock cycles after the output is disabled (PCLOEn = 0), the PCLBUZn output width becomes shorter.

CHAPTER 10 WATCHDOG TIMER

10.1 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 17 RESET FUNCTION**.

When 75% + 1/2f_{IL} of the overflow time is reached, an interval interrupt can be generated.

10.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 10-1. Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

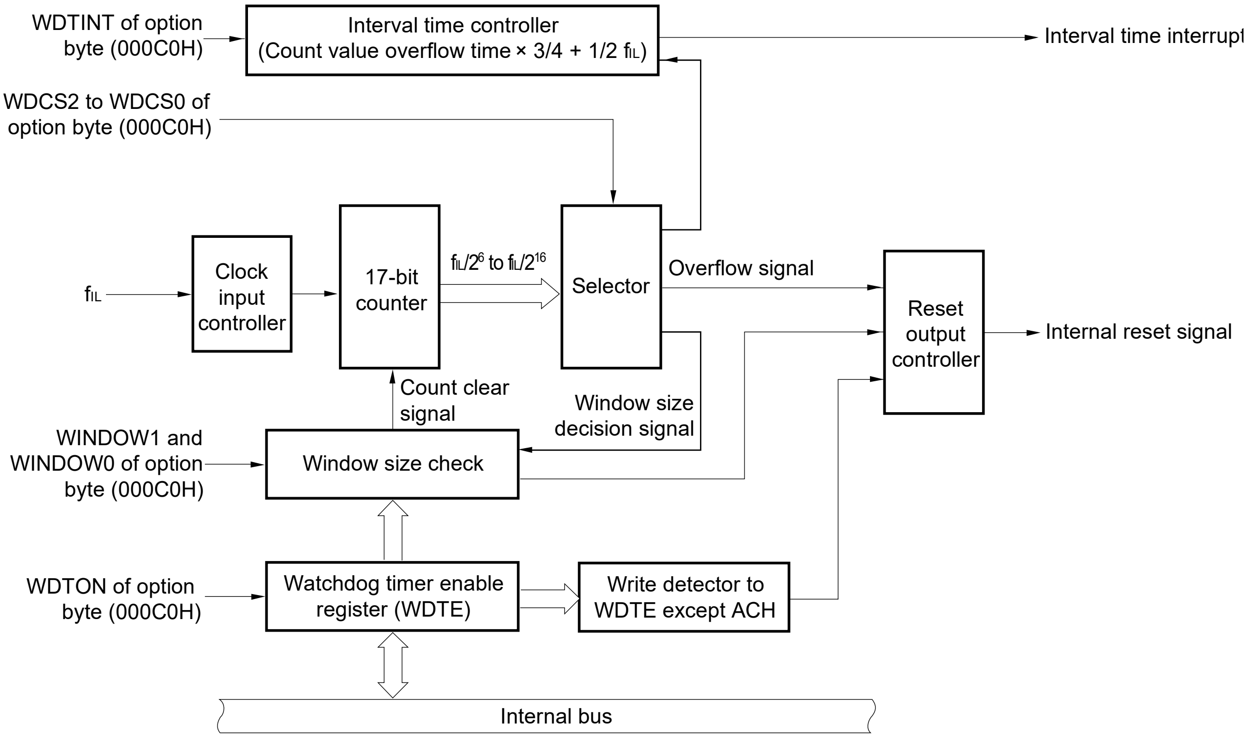
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 10-2. 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 22 OPTION BYTE**.

Figure 10-1. Block Diagram of Watchdog Timer



Remark f_{IL}: Low-speed on-chip oscillator clock frequency

10.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

(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 10-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH After reset: 9AH/1AHNote R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Cautions 1. If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.

2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.

3. The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).

10.4 Operation of Watchdog Timer

10.4.1 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 22**).

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 **10.4.2** and **CHAPTER 22**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **10.4.3** and **CHAPTER 22**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write the WDTE register the second time or later after a reset release during the window open period. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to the WDTE register, an internal reset signal is generated.
   An internal reset signal is generated in the following cases.

   - If a 1-bit manipulation instruction is executed on the WDTE register
   - If data other than “ACH” is written to the WDTE register

Cautions 1. When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

2. After "ACH" is written to the WDTE register, an error of up to 2 clocks (f_{IL}) may occur before the watchdog timer counter is cleared.

3. The watchdog timer can be cleared immediately before the count value overflows.

Cautions 4. The operation of the watchdog timer in the HALT, STOP, and SNOOZE modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		
In SNOOZE mode		

If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts.
When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

10.4.2 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 10-3. 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 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.

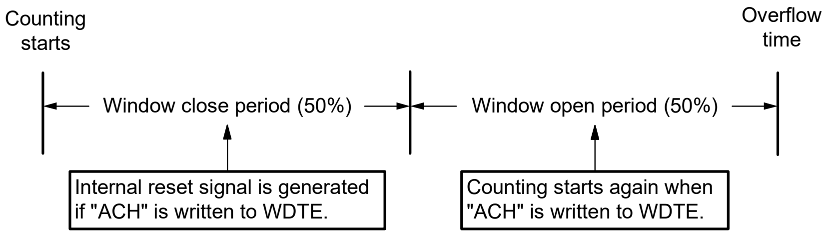
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

10.4.3 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 10-4. 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

10.4.4 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 10-5. 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 11 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

	48-pin	64-pin
Analog input channels	9 ch (ANI0, ANI1, ANI16 to ANI22)	10 ch (ANI0, ANI1, ANI16 to ANI23)

Caution Most of the following descriptions in this chapter use the 64-pin as an example.

11.1 Function of A/D Converter

The A/D converter is used to convert analog input signals into digital values, and is configured to control analog inputs, including up to 10 channels of A/D converter analog inputs (ANI0, ANI1 and ANI16 to ANI23). 10-bit or 8-bit resolution can be selected by the ADTYP bit of the A/D converter mode register 2 (ADM2).

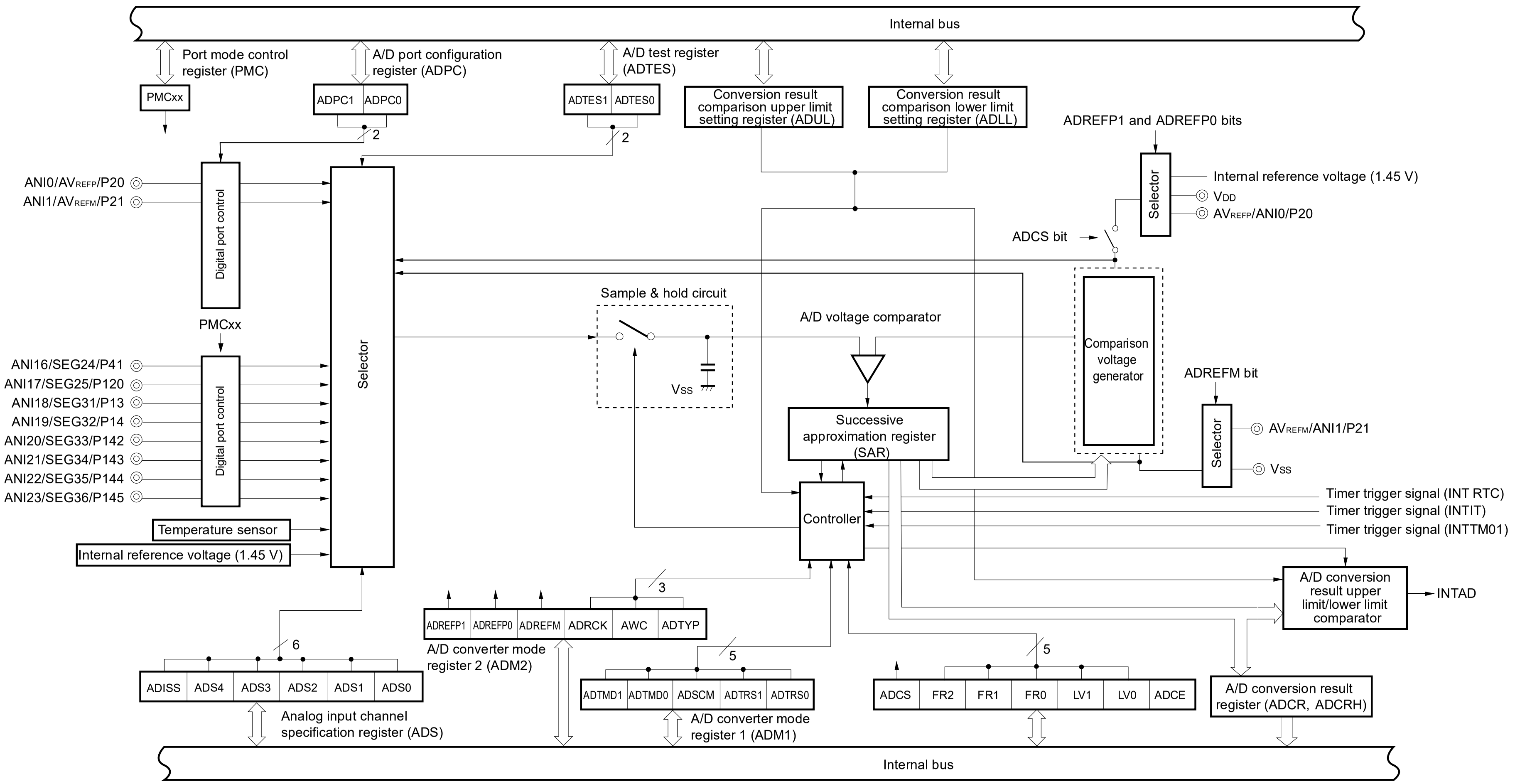
The A/D converter has the following function.

- **10-bit resolution A/D conversion**
  10-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0, ANI1 and ANI16 to ANI23. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger	Conversion is started by software.
	Hardware trigger no-wait mode	Conversion is started by detecting a hardware trigger.
	Hardware trigger wait mode	The power is turned on by detecting a hardware trigger while the system is off and in the conversion standby state, and conversion is then started automatically after the stabilization wait time passes. When using the SNOOZE mode function, specify the hardware trigger wait mode.
Conversion operation mode	One-shot conversion mode	A/D conversion is performed on the selected channel once.
	Sequential conversion mode	A/D conversion is sequentially performed on the selected channels until it is stopped by software.
Operation voltage mode	Standard 1 or standard 2 mode	Conversion is done in the operation voltage range of $2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$.
	Low voltage 1 or low voltage 2 mode	Conversion is done in the operation voltage range of $1.6 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$. Select this mode for conversion at a low voltage. Because the operation voltage is low, it is internally boosted during conversion.
Sampling time selection	Sampling clock cycles: 7 f_{AD}	The sampling time in standard 1 or low voltage 1 mode is seven cycles of the conversion clock (f_{AD}). Select this mode when the output impedance of the analog input source is high and the sampling time should be long.
	Sampling clock cycles: 5 f_{AD}	The sampling time in standard 2 or low voltage 2 mode is five cycles of the conversion clock (f_{AD}). Select this mode when enough sampling time is ensured (for example, when the output impedance of the analog input source is low).

Figure 11-1. Block Diagram of A/D Converter



Remark Analog input pin for figure 11-1 when a 64-pin product is used.

11.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0, ANI1 and ANI16 to ANI23 pins

These are the analog input pins of the 10 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 the voltage generated from the voltage tap of the comparison voltage generator with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REF}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AV_{REF}), the MSB bit of the SAR is reset.

After that, bit 8 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the comparison voltage generator is selected by the value of bit 9, to which the result has been already set.

Bit 9 = 0: (1/4 AV_{REF})
Bit 9 = 1: (3/4 AV_{REF})

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 8 = 1
Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.

When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 2 of the SAR register.

Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

(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) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(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 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 ANI16 to ANI23 are converted to digital signals based on the voltage applied between AV_{REFP} and the – side reference voltage (AV_{REFM}/V_{SS}).

In addition to AV_{REFP}, it is possible to select V_{DD} or the internal reference voltage (1.45 V) as the + side reference voltage of the A/D converter.

(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 V_{SS} as the – side reference voltage of the A/D converter.

11.3 Registers Used in 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)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- A/D port configuration register (ADPC)
- Port mode control registers 1, 4, 12, and 14 (PMC1, PMC4, PMC12, PMC14)
- Port mode registers 1, 2, 4, 12, and 14 (PM1, PM2, PM4, PM12, PM14)

11.3.1 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 11-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

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

Cautions 1. When setting the A/D converter, be sure to set the following registers first while the ADCEN bit is set to 1. If ADCEN = 0, writing to a control register of the A/D converter is ignored, and, even if the register is read, only the default value is read (except for port mode registers 1, 2, 4, 12, and 14 (PM1, PM2, PM4, PM12, and PM14), port mode registers 1, 2, 4, 12, and 14 (PM1, PM2, PM4, PM12, and PM14), port mode control registers 1, 4, 12, and 14 (PMC1, PMC4, PMC12, PMC14), and A/D port configuration register (ADPC)).

- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 1 (ADM1)**
- **A/D converter mode register 2 (ADM2)**
- **10-bit A/D conversion result register (ADCR)**
- **8-bit A/D conversion result register (ADCRH)**
- **Analog input channel specification register (ADS)**
- **Conversion result comparison upper limit setting register (ADUL)**
- **Conversion result comparison lower limit setting register (ADLL)**
- **A/D test register (ADTES).**

2. Be sure to clear bits 1, 3, 4 and 6 to 0.

11.3.2 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 11-3. 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	0	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: Stabilization wait status + conversion operation status

ADCE	A/D voltage comparator operation control[Note 2]
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

Notes

1. For details of the FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Table 11-3 A/D Conversion Time Selection**.
2. In software trigger mode and hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and time is required for the conversion value to stabilize after the A/D converter starts operating (1.0 μs). Valid conversion results can therefore be obtained from the first conversion by setting the ADCE bit to 1 and then waiting for the stabilization time (1.0 μs) to elapse before setting the ADCS bit to 1. If the ADCS bit is set to 1 before the stabilization time (1.0 μs) elapses, the first conversion data must be ignored.

Cautions

1. **Change the FR2 to FR0, LV1, and LV0 bits while conversion is stopped (ADCS = 0, ADCE = 0).**
2. **Do not set the ADCS bit to 1 and the ADCE bit to 0 at the same time.**
3. **Do not change the ADCS and ADCE bits from 0 to 1 at the same time by using an 8-bit manipulation instruction. Be sure to set these bits in the order described in 11.7 A/D Converter Setup Flowchart.**
4. **Be sure to clear bit 6 to 0.**

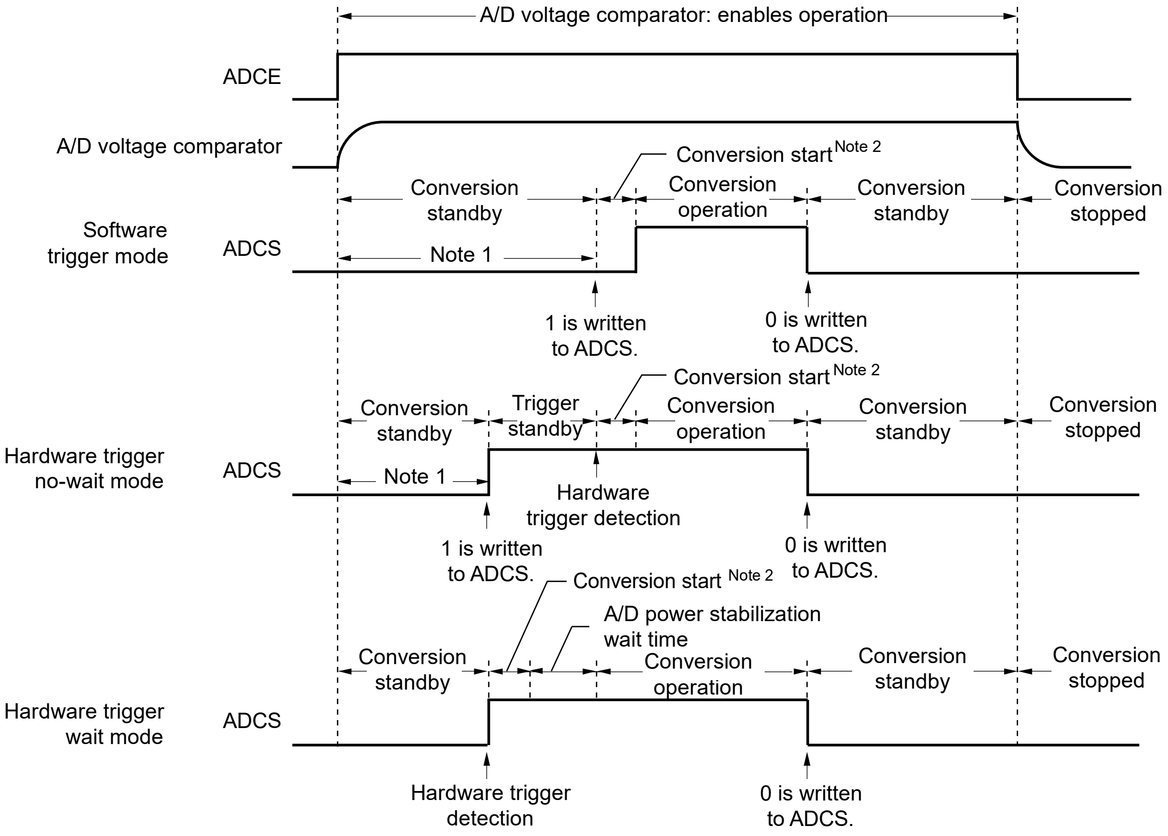
Table 11-1. Settings of ADCS and ADCE Bits

ADCS	ADCE	A/D Conversion Operation
0	0	Stop status
0	1	Conversion standby mode
1	0	Setting prohibited
1	1	Conversion mode

Table 11-2. Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode		Set Conditions	Clear Conditions
Software trigger mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
	One-shot conversion mode		● When 0 is written to ADCS ● The bit is automatically cleared to 0 when A/D conversion ends.
Hardware trigger no-wait mode	Sequential conversion mode		When 0 is written to ADCS
	One-shot conversion mode		When 0 is written to ADCS
Hardware trigger wait mode	Sequential conversion mode	When a hardware trigger is input	When 0 is written to ADCS
	One-shot conversion mode		● When 0 is written to ADCS ● The bit is automatically cleared to 0 when A/D conversion ends.

Figure 11-4. Timing Chart When A/D Voltage Comparator Is Used



Notes 1. While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer to stabilize the internal circuit.

2. The following time is the maximum amount of time necessary to start conversion.

ADM0			Conversion Clock (f_{AD})	Conversion Start Time (Number of f_{CLK} Clocks)	
FR2	FR1	FR0		Software trigger mode/ Hardware trigger no wait mode	Hardware trigger wait mode
0	0	0	f_{CLK}/64	63	1
0	0	1	f_{CLK}/32	31	
0	1	0	f_{CLK}/16	15	
0	1	1	f_{CLK}/8	7	
1	0	0	f_{CLK}/6	5	
1	0	1	f_{CLK}/5	4	
1	1	0	f_{CLK}/4	3	
1	1	1	f_{CLK}/2	1	

However, for the second and subsequent conversion in sequential conversion mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

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

Cautions 1. If using the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (but the bit is automatically switched to 1 when the hardware trigger signal is detected). However, it is possible to clear the ADCS bit to 0 to specify the A/D conversion standby status.

2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS flag is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.

3 Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby status).

4. To complete A/D conversion, specify at least the following time as the hardware trigger interval:

Hardware trigger no wait mode:	**2 f_{CLK} clock + conversion start time + A/D conversion time**
Hardware trigger wait mode:	**2 f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time**

Table 11-3. A/D Conversion Time Selection (1/4)

(1) When there is no stabilization wait time

Normal mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock Cycles Note 3	Conversion Time	Conversion Time Selection				
									2.7 V ≤ V_{DD} ≤ 5.5 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}/64	19 f_{AD} (number of sampling clock cycles: 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	76 µs	50.6667 µs
0	0	1				f_{CLK}/32		608/f_{CLK}			76 µs	38 µs	25.3333 µs
0	1	0				f_{CLK}/16		304/f_{CLK}		76 µs	38 µs	19 µs	12.6667 µs
0	1	1				f_{CLK}/8		152/f_{CLK}		38 µs	19 µs	9.5 µs	6.3333 µs
1	0	0				f_{CLK}/6		114/f_{CLK}		28.5 µs	14.25 µs	7.125 µs	4.75 µs
1	0	1				f_{CLK}/5		95/f_{CLK}	95 µs	23.75 µs	11.875 µs	5.938 µs	3.9583 µs
1	1	0				f_{CLK}/4		76/f_{CLK}	76 µs	19 µs	9.5 µs	4.75 µs	3.1667 µs Note 1
1	1	1				f_{CLK}/2		38/f_{CLK}	38 µs	9.5 µs	4.75 µs	2.375 µs Notes 1, 2	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	17 f_{AD} (number of sampling clock cycles: 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	68 µs	45.3333 µs
0	0	1				f_{CLK}/32		544/f_{CLK}			68 µs	34 µs	22.6667 µs
0	1	0				f_{CLK}/16		272/f_{CLK}		68 µs	34 µs	17 µs	11.3333 µs
0	1	1				f_{CLK}/8		136/f_{CLK}		34 µs	17 µs	8.5 µs	5.6667 µs
1	0	0				f_{CLK}/6		102/f_{CLK}		25.5 µs	12.75 µs	6.375 µs	4.25 µs
1	0	1				f_{CLK}/5		85/f_{CLK}	85 µs	21.25 µs	10.625 µs	5.3125 µs	3.5417 µs
1	1	0				f_{CLK}/4		68/f_{CLK}	68 µs	17 µs	8.5 µs	4.25 µs	2.8333 µs Notes 1, 2
1	1	1				f_{CLK}/2		34/f_{CLK}	34 µs	8.5 µs	4.25 µs	2.125 µs Notes 1, 2	Setting prohibited

Notes 1. Setting prohibited when V_{DD} < 3.6 V.

2. This value is prohibited when using the temperature sensor.

3. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 27.6.1 A/D converter characteristics.

2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, make sure that conversion has stopped (ADCS = 0, ADCE = 0).

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (2/4)

(2) When there is no stabilization wait time Note 1

Low-voltage mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock Cycles Note 8	Conversion Time	Conversion Time Selection				
FR2	FR1	FR0	LV1	LV0					1.6 V ≤ V_{DD} ≤ 5.5 V		Note 2	Note 3	Note 4
									f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	1	0	Low-voltage 1	f_{CLK}/64	19 f_{AD} (number of sampling clock cycles: 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	76 µs	50.6667 µs
0	0	1				f_{CLK}/32		608/f_{CLK}			76 µs	38 µs	25.3333 µs
0	1	0				f_{CLK}/16		304/f_{CLK}		76 µs	38 µs	19 µs	12.6667 µs
0	1	1				f_{CLK}/8		152/f_{CLK}		38 µs Note 7	19 µs	9.5 µs Note 6	6.3333 µs
1	0	0				f_{CLK}/6		114/f_{CLK}		28.5 µs Note 7	14.25 µs Note 6	7.125 µs Note 6	4.75 µs
1	0	1				f_{CLK}/5		95/f_{CLK}	95 µs	23.75 µs Note 7	11.875 µs Note 6	5.938 µs Note 6	3.9587 µs
1	1	0				f_{CLK}/4		76/f_{CLK}	76 µs	19 µs Note 7	9.5 µs Note 6	4.75 µs Note 6	3.1667 µs Note 5
1	1	1				f_{CLK}/2		38/f_{CLK}	38 µs Note 7	9.5 µs Note 6	4.75 µs Note 6	2.375 µs Note 5	Setting prohibited
0	0	0	1	1	Low-voltage 2	f_{CLK}/64	17 f_{AD} (number of sampling clock cycles: 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	68 µs	45.3333 µs
0	0	1				f_{CLK}/32		544/f_{CLK}			68 µs	34 µs	22.6667 µs
0	1	0				f_{CLK}/16		272/f_{CLK}		68 µs	34 µs	17 µs	11.3333 µs
0	1	1				f_{CLK}/8		136/f_{CLK}		34 µs Note 7	17 µs	8.5 µs Note 6	5.6667 µs
1	0	0				f_{CLK}/6		102/f_{CLK}		25.5 µs Note 7	12.75 µs Note 6	6.375 µs Note 6	4.25 µs
1	0	1				f_{CLK}/5		85/f_{CLK}	85 µs	21.25 µs Note 7	10.625 µs Note 6	5.3125 µs Note 6	3.5417 µs
1	1	0				f_{CLK}/4		68/f_{CLK}	68 µs	17 µs Note 7	8.5 µs Note 6	4.25 µs Note 6	2.8333 µs Note 5
1	1	1				f_{CLK}/2		34/f_{CLK}	34 µs Note 7	8.5 µs Note 6	4.25 µs Note 6	2.125 µs Note 5	Setting prohibited

Notes 1. This mode is prohibited when using the temperature sensors.

2. 1.8 V ≤ V_{DD} ≤ 5.5 V

3. 2.4 V ≤ V_{DD} ≤ 5.5 V

4. 2.7 V ≤ V_{DD} ≤ 5.5 V

5. Setting prohibited when V_{DD} < 3.6 V.

6. Setting prohibited when V_{DD} < 2.7 V.

7. Setting prohibited when V_{DD}< 1.8 V.

8. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 27.6.1 A/D converter characteristics.

2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, make sure that conversion has stopped (ADCS = 0, ADCE = 0).

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (3/4)

(3) When there is stabilization wait time
Normal mode 1, 2 (hardware trigger wait mode $^{\text{Note 1}}$)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Stabilization Wait Cock	Number of Conversion Clock Cycles Note 4	Stabilization Wait Cock + Conversion Time	Conversion Time Selection				
										2.7 V ≤ V_{DD} ≤ 5.5 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}/64	8 f_{AD}	19 f_{AD} (number of sampling clock cycles: 7 f_{AD})	1728/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	108 μs	72 μs
0	0	1				f_{CLK}/32			864/f_{CLK}			108 μs	54 μs	36 μs
0	1	0				f_{CLK}/16			432/f_{CLK}		108 μs	54 μs	27 μs	18 μs
0	1	1				f_{CLK}/8			216/f_{CLK}		54 μs	27 μs	13.5 μs	9 μs
1	0	0				f_{CLK}/6			162/f_{CLK}		40.5 μs	20.25 μs	10.125 μs	6.75 μs
1	0	1				f_{CLK}/5			135/f_{CLK}	135 μs	33.75 μs	16.875 μs	8.4375 μs	5.625 μs
1	1	0				f_{CLK}/4			108/f_{CLK}	108 μs	27 μs	13.5 μs	6.75 μs	4.5 μs
1	1	1				f_{CLK}/2			54/f_{CLK}	54 μs	13.5 μs	6.75 μs	3.375 μs Notes 3	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	8 f_{AD}	17 f_{AD} (number of sampling clock cycles: 5 f_{AD})	1600/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	100 μs	66.6667 μs
0	0	1				f_{CLK}/32			800/f_{CLK}			100 μs	50 μs	33.3333 μs
0	1	0				f_{CLK}/16			400/f_{CLK}		100 μs	50 μs	25 μs	16.6667 μs
0	1	1				f_{CLK}/8			200/f_{CLK}		50 μs	25 μs	12.5 μs	8.3333 μs
1	0	0				f_{CLK}/6			150/f_{CLK}		37.5 μs	18.75 μs	9.375 μs	6.25 μs
1	0	1				f_{CLK}/5			125/f_{CLK}	125 μs	31.25 μs	15.625 μs	7.8125 μs	5.2083 μs
1	1	0				f_{CLK}/4			100/f_{CLK}	100 μs	25 μs	12.5 μs	6.25 μs	4.1667 μs Notes 2, 3
1	1	1				f_{CLK}/2			50/f_{CLK}	50 μs	12.5 μs	6.25 μs	3.125 μs Notes 2, 3	Setting prohibited

Notes 1. For the second and subsequent conversion in sequential conversion 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 11-3 (1/4)**).

2. Setting prohibited when V_{DD} < 3.6 V.

3. This value is prohibited when using the temperature sensors.

4. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 27.6.1 A/D converter characteristics.

2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, make sure that conversion has stopped (ADCS = 0, ADCE = 0).

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

4. When hardware trigger wait mode, specify the conversion time, including the stabilization wait time from the hardware trigger detection.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (4/4)

(4) When there is no stabilization wait time

Low-voltage mode 1, 2 Note 1 (hardware trigger wait mode Note 2)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Stabilization Wait Cock	Number of Conversion Clock Cycles Note 9	Stabilization Wait Cock + Conversion Time	Conversion Time Selection				
										1.6 V ≤ V_{DD} ≤ 5.5 V		Note 3	Note 4	Note 5
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	Low voltage 1	f_{CLK}/64	2 f_{AD}	19 f_{AD} (number of sampling clock cycles: 7 f_{AD})	1344/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	84 μs	56 μs
0	0	1				f_{CLK}/32			672/f_{CLK}			84 μs	42 μs	28 μs
0	1	0				f_{CLK}/16			336/f_{CLK}		84 μs	42 μs	21 μs	14 μs
0	1	1				f_{CLK}/8			168/f_{CLK}		42 μs Note 8	21 μs	10.5 μs Note 7	7 μs
1	0	0				f_{CLK}/6			126/f_{CLK}		31.25 μs Note 8	15.75 μs Note 7	7.875 μs Note 7	5.25 μs
1	0	1				f_{CLK}/5			105/f_{CLK}	105 μs	26.25 μs Note 8	13.125 μs Note 7	6.5625 μs Note 7	4.375 μs
1	1	0				f_{CLK}/4			84/f_{CLK}	84 μs	21 μs Note 8	10.5 μs Note 7	5.25 μs Note 7	3.5 μs Note 6
1	1	1				f_{CLK}/2			42/f_{CLK}	42 μs Note 8	10.5 μs Note 7	5.25 μs Note 7	2.625 μs Note 6	Setting prohibited
0	0	0	0	1	Low voltage 2	f_{CLK}/64	2 f_{AD}	17 f_{AD} (number of sampling clock cycles: 5 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	76 μs	50.6667 μs
0	0	1				f_{CLK}/32			608/f_{CLK}			76 μs	38 μs	25.3333 μs
0	1	0				f_{CLK}/16			304/f_{CLK}		76 μs	38 μs	19 μs	12.6667 μs
0	1	1				f_{CLK}/8			152/f_{CLK}		38 μs Note 8	19 μs	9.5 μs Note 7	6.3333 μs
1	0	0				f_{CLK}/6			114/f_{CLK}		28.5 μs Note 8	14.25 μs Note 7	7.125 μs Note7	4.75 μs
1	0	1				f_{CLK}/5			96/f_{CLK}	96 μs	23.75 μs Note 8	12 μs Note 7	5.938 μs Note 7	4.0 μs
1	1	0				f_{CLK}/4			76/f_{CLK}	76 μs	19 μs Note 8	9.5 μs Note 7	4.75 μs Note 7	3.1667 μs Note 6
1	1	1				f_{CLK}/2			38/f_{CLK}	38 μs Note 8	9.5 μs Note 7	4.75 μs Note 7	2.375 μs Note 6	Setting prohibited

Notes 1. This mode is prohibited when using the temperature sensor

2. For the second and subsequent conversion in sequential conversion 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 11-3 (2/4)**).

3. 1.8 V ≤ V_{DD} ≤ 5.5 V

4. 2.4 V ≤ V_{DD} ≤ 5.5 V

5. 2.7 V ≤ V_{DD} ≤ 5.5 V

6. Setting prohibited when V_{DD} < 3.6 V.

7. Setting prohibited when V_{DD}< 2.7 V.

8. Setting prohibited when V_{DD} < 1.8 V.

9. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

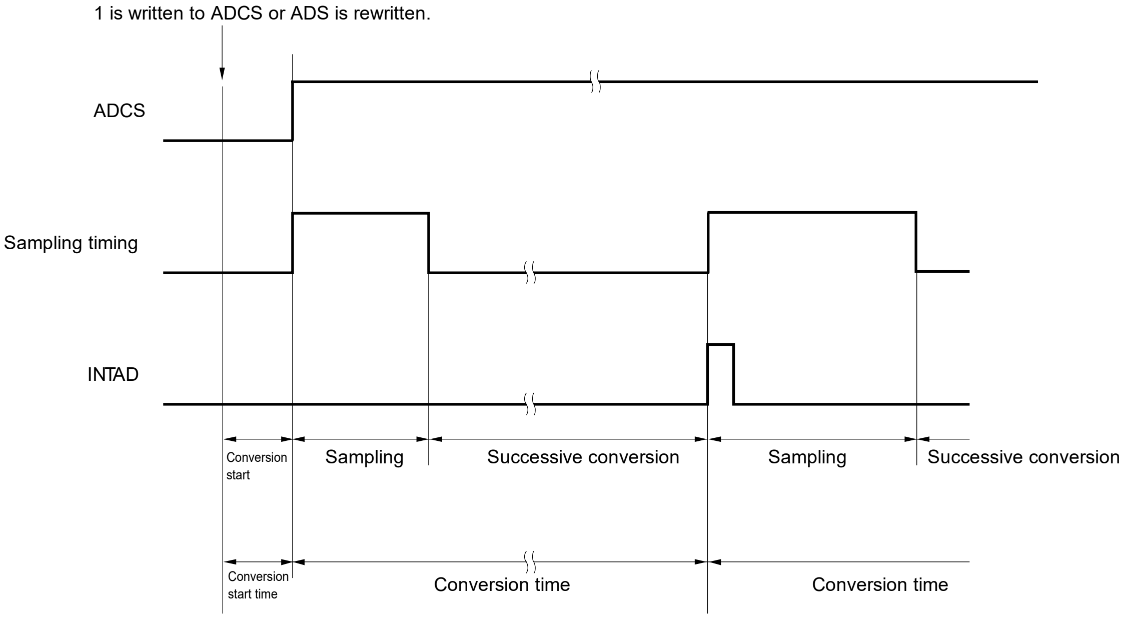
Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 27.6.1 A/D converter characteristics.

2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, make sure that conversion has stopped (ADCS = 0, ADCE = 0).

3. **The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.**
4. **When hardware trigger wait mode, specify the conversion time, including the stabilization wait time from the hardware trigger detection.**

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 11-5. A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



11.3.3 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 11-6. 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	Setting prohibited
1	0	Real-time clock interrupt signal (INTRTC)
1	1	12-bit interval timer interrupt signal (INTIT)

Cautions 1. Rewrite the value of the ADM1 register while conversion is stopped (ADCS = 0, ADCE = 0).

2. To complete A/D conversion, specify at least the following time as the hardware trigger interval:

Hardware trigger no wait mode: 2 f_{CLK} clock + conversion start time + A/D conversion time

Hardware trigger wait mode: 2 f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time

3. In modes other than SNOOZE mode, input of the next INTRTC or INTIT will not be recognized as a valid hardware trigger for up to four f_{CLK} cycles after the first INTRTC or INTIT is input.

Remarks 1. **×**: don't care

2. f_{CLK}: CPU/peripheral hardware clock frequency

11.3.4 A/D converter mode register 2 (ADM2)

This register is used to select the + side or - side reference voltage of the A/D converter, check the upper limit and lower limit A/D conversion result values, select the resolution, and specify whether to use the SNOOZE mode.

The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-7. Format of A/D Converter Mode Register 2 (ADM2) (1/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	Selection of the + side reference voltage source of the A/D converter
0	0	Supplied from V_{DD}
0	1	Supplied from P20/AV_{REFP}/ANI0
1	0	Supplied from the internal reference voltage (1.45 V) Note
1	1	Setting prohibited

- When ADREFP1 or ADREFP0 bit is rewritten, this must be configured in accordance with the following procedures.
  (1) Set ADCE = 0
  (2) Change the values of ADREFP1 and ADREFP0
  (3) Stabilization wait time (A)
  (4) Set ADCE = 1
  (5) Stabilization wait time (B)
  When ADREFP1 and ADREFP0 are set to 1 and 0, the setting is changed to A = 5 μs, B = 1 μs.
  When ADREFP1 and ADREFP0 are set to 0 and 0 or 0 and 1, A needs no wait and B = 1 μs.
- When ADREFP1 and ADREFP0 are set to 1 and 0, respectively, A/D conversion cannot be performed on the temperature sensor output voltage and internal reference voltage.
  Be sure to perform A/D conversion while ADISS = 0.

ADREFM	Selection of the – side reference voltage source of the A/D converter
0	Supplied from V_{SS}
1	Supplied from P21/AV_{REFM}/ANI1

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (AREA 1).
1	The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (AREA 2) or the ADUL register < the ADCR register (AREA 3).
Figure 11-8 shows the generation range of the interrupt signal (INTAD) for <AREA 1> to <AREA 3>.	

Note This setting can be used only in HS (high-speed main) mode.

Cautions 1. Rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

2. Do not set the ADREFP1 bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock. Also, if the ADREFP1 bit is set to 1, the temperature sensor operating current (I_{ADREF}) indicated in 27.3.2 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.

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 11-7. Format of A/D Converter Mode Register 2 (ADM2) (2/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.

When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode).

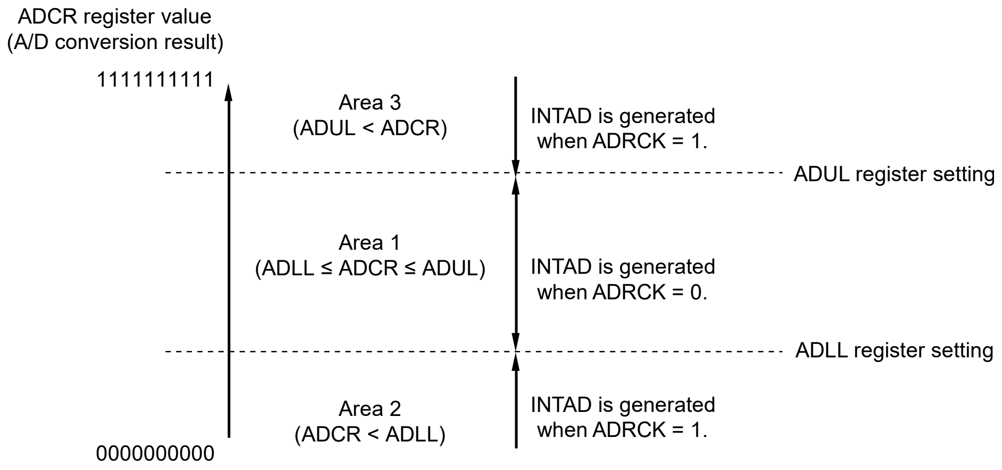
- The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited.
- Using the SNOOZE mode function in the sequential conversion mode is prohibited.
- When using the SNOOZE mode function, specify a hardware trigger interval of at least “shift time to SNOOZE mode Note + conversion start time + A/D power supply stabilization wait time + A/D conversion time +2 f_{CLK} clock”
- Even when using SNOOZE mode, be sure to set the AWC bit to 0 in normal operation mode 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 mode.
  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.

ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

Note Refer to “Transition time from STOP mode to SNOOZE mode” in **16.3.3 SNOOZE mode**.

Caution Only rewrite the value of the ADM2 register while conversion operation is stopped (which is indicated by the ADCS bit of A/D converter mode register 0 (ADM0) being 0).

Figure 11-8. ADRCK Bit Interrupt Signal Generation Range



Remark If INTAD does not occur, the A/D conversion result is not stored in the ADCR or ADCRH register.

11.3.5 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH **Note**.

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 11-8**), the result is not stored.

Figure 11-9. Format of 10-bit A/D Conversion Result Register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR											0	0	0	0	0	0

Cautions 1. When 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.

2. When 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1) and the ADCR register is read, 0 is read from the lower two bits (ADCR1 and ADCR0).

3. When the ADCR register is accessed in 16-bit units, the higher 10 bits of the conversion result are read in order starting at bit 15.

11.3.6 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored [Note].

The ADCRH register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 11-8**), the result is not stored.

Figure 11-10. Format of 8-bit A/D Conversion Result Register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of the ADCRH register may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, and ADPC registers. Using timing other than the above may cause an incorrect conversion result to be read.

11.3.7 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 11-11. Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AV$_{REFP}$ pin
0	0	0	0	0	1	ANI1	P21/ANI1/AV$_{REFM}$ pin
0	1	0	0	0	0	ANI16	P41/ANI16 pin
0	1	0	0	0	1	ANI17	P120/ANI17 pin
0	1	0	0	1	0	ANI18	P13/ANI18 pin
0	1	0	0	1	1	ANI19	P14/ANI19 pin
0	1	0	1	0	0	ANI20	P142/ANI20 pin
0	1	0	1	0	1	ANI21	P143/ANI21 pin
0	1	0	1	1	0	ANI22	P144/ANI22 pin
0	1	0	1	1	1	ANI23	P145/ANI23 pin
1	0	0	0	0	0	–	Temperature sensor output voltage Note
1	0	0	0	0	1	–	Internal reference voltage (1.45 V) Note
Other than the above						Setting prohibited	

Note Can only be used in HS (high-speed main) mode.

Cautions 1. Be sure to clear bits 5 and 6 to 0.

2. Set a channel to be set the analog input by ADPC and PMC registers in the input mode by using port mode registers 1, 2, 4, 12, and 14 (PM1, PM2, PM4, PM12, PM14).

3. Do not set the pin that is set by the A/D port configuration register (ADPC) as digital I/O by the ADS register.

4. Do not set the pin that is set by port mode control register 1, 4, 12, or 14 (PMC1, PMC4, PMC12, PMC14) as digital I/O by the ADS register.

5. Rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

6. If using AV$_{REFP}$ as the + side reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.

7. If using AV$_{REFM}$ as the – side reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.

8. If ADISS is set to 1, the internal reference voltage (1.45 V) cannot be used for the + side reference voltage source. After the ADISS bit is set to 1, the initial conversion result cannot be used.

For the setting flow, see 11.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode).

Cautions 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 ADREFP1 bit is set to 1, the A/D converter reference voltage current (I_{ADREF}) indicated in 27.3.2 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.

11.3.8 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 11-8**).

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

Reset signal generation sets this register to FFH.

Caution When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADUL register.

Figure 11-12. Format of Conversion Result Comparison Upper Limit Setting Register (ADUL)

Address: F0011H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

11.3.9 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 11-8**).

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

Reset signal generation clears this register to 00H.

Figure 11-13. Format of Conversion Result Comparison Lower Limit Setting Register (ADLL)

Address: F0012H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Cautions 1. When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADLL register.

2. Only write new values to the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).

3. The setting of the ADUL registers must be greater than that of the ADLL register.

11.3.10 A/D test register (ADTES)

This register is used to select the + side reference voltage or - side reference voltage for the converter, an analog input channel (ANIxx), the temperature sensor output voltage, or the internal reference voltage (1.45 V) as the target for A/D conversion.

When using this register to test the converter, set as follows.

- For zero-scale measurement, select the - side reference voltage as the target for conversion.
- For full-scale measurement, select the + side reference voltage as the target for conversion.

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

Reset signal generation clears this register to 00H.

Figure 11-14. Format of A/D Test Register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx/temperature sensor output voltage [Note]/internal reference voltage (1.45 V)[Note] (This is specified using the analog input channel specification register (ADS).)
1	0	The - side reference voltage (selected by the ADREFM bit of the ADM2 register)
1	1	The + side reference voltage (selected by the ADREFP1 or ADREFP0 bit of the ADM2 register)
Other than the above		Setting prohibited

Note The temperature sensor output voltage and internal reference voltage (1.45 V) can be selected only in the HS (high-speed main) mode.

11.3.11 Registers controlling port function of analog input pins

Set up the registers for controlling the functions of the ports shared with the analog input pins of the A/D converter (port mode registers (PMxx), port mode control registers (PMCxx), and A/D port configuration register (ADPC)). For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.6 Port mode control registers (PMCxx)**, and **4.3.7 A/D port configuration register (ADPC)**.

When using the ANI0 and ANI1 pins for analog input of the A/D converter, set the port mode register (PMxx) bit corresponding to each port to 1 and select analog input through the A/D port configuration register (ADPC).

When using the ANI16 to ANI23 pins for analog input of the A/D converter, set the port mode register (PMxx) bit and port mode control register (PMCxx) bit corresponding to each port to 1.

11.4 A/D Converter Conversion Operations

The A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<2> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AVREF by the tap selector.

<4> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than (1/2) AVREF, the MSB bit of the SAR register remains set to 1. If the analog input is smaller than (1/2) AVREF, the MSB bit is reset to 0.

<5> Next, bit 8 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) AVREF
- Bit 9 = 0: (1/4) AVREF

The voltage tap and sampled voltage are compared and bit 8 of the SAR register is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison of 10 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched [Note 1].
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated [Note 1].

<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0 [Note 2].
To stop the A/D converter, clear the ADCS bit to 0.

Notes 1. If the A/D conversion result is outside the A/D conversion result range specified by the ADRCK bit and the ADUL and ADLL registers (see **Figure 11-8**), the A/D conversion result interrupt request signal is not generated and no A/D conversion results are stored in the ADCR and ADCRH registers.

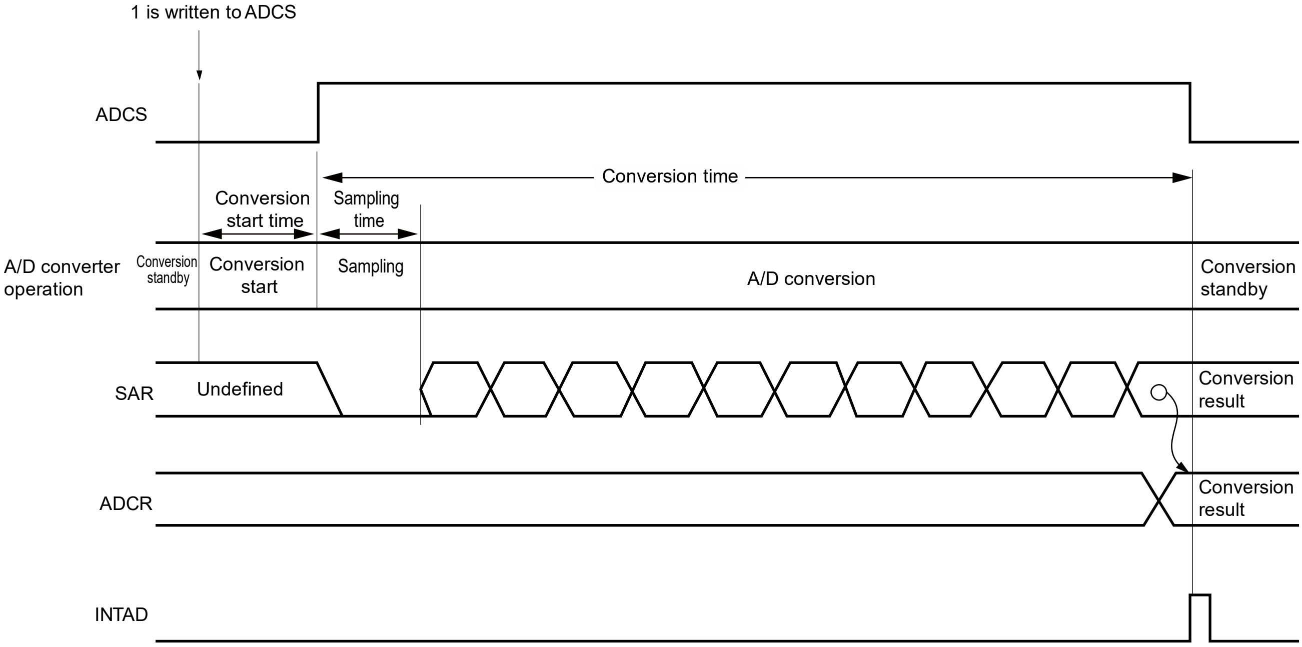
2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

Remarks 1. Two types of the A/D conversion result registers are available.

● ADCR register (16 bits):	Store 10-bit A/D conversion value
● ADCRH register (8 bits):	Store 8-bit A/D conversion value

2. AVREF: The + side reference voltage of the A/D converter. This can be selected from AVREFP, the internal reference voltage (1.45 V), and VDD.

Figure 11-15. Conversion Operation of A/D Converter (Software Trigger Mode)



In one-shot conversion mode, the ADCS bit is automatically cleared to 0 after completion of A/D conversion.

In sequential conversion mode, A/D conversion operations proceed continuously until the software clears bit 7 (ADCS) of the A/D converter mode register 0 (ADM0) to 0.

Writing to the analog input channel specification register (ADS) during A/D conversion interrupts the current conversion after which A/D conversion of the analog input specified by the ADS register proceeds. Data from the A/D conversion that was in progress are discarded.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

11.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0, ANI1, ANI16 to ANI23) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$\mathrm{SAR} = \mathrm{INT}\,\left(\frac{V_{\mathrm{AIN}}}{AV_{\mathrm{REF}}} \times 1024 + 0.5\right)$$

$$\mathrm{ADCR} = \mathrm{SAR} \times 64$$

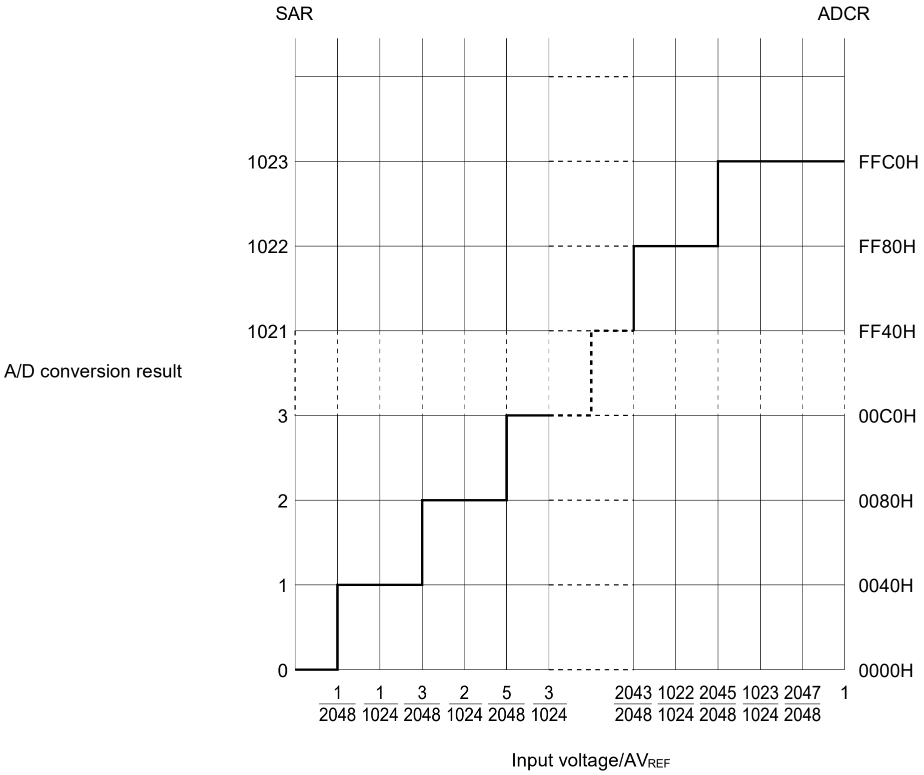
or

$$\left(\frac{\mathrm{ADCR}}{64} - 0.5\right) \times \frac{AV_{\mathrm{REF}}}{1024} \leq V_{\mathrm{AIN}} < \left(\frac{\mathrm{ADCR}}{64} + 0.5\right) \times \frac{AV_{\mathrm{REF}}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses
V_{AIN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR: A/D conversion result register (ADCR) value
SAR: Successive approximation register

Figure 11-16 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 11-16. Relationship Between Analog Input Voltage and A/D Conversion Result



Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

11.6 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 **11.7 A/D Converter Setup Flowchart**.

11.6.1 Software trigger mode (sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1.0 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

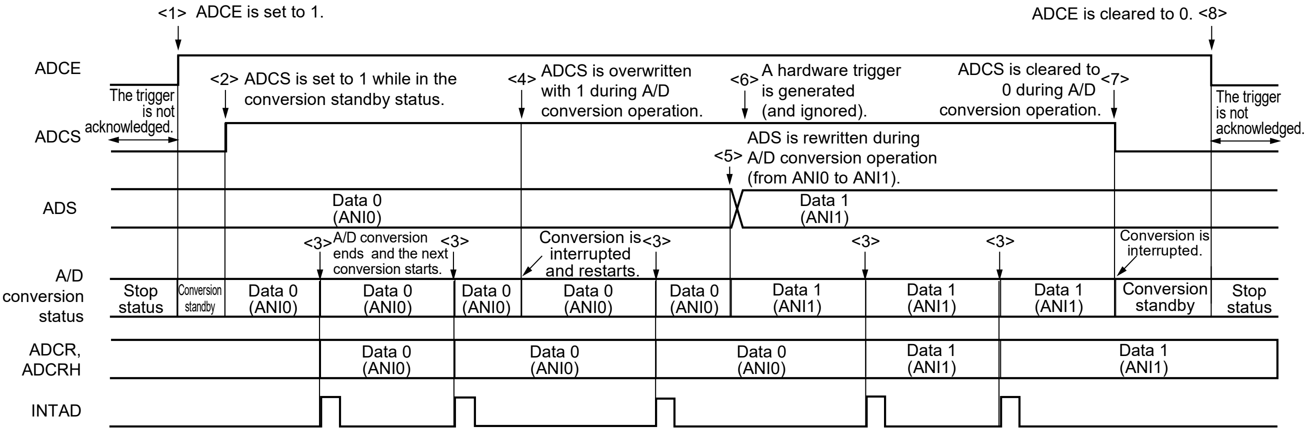
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 11-17. Example of Software Trigger Mode (Sequential Conversion Mode) Operation Timing



11.6.2 Software trigger mode (one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1.0 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

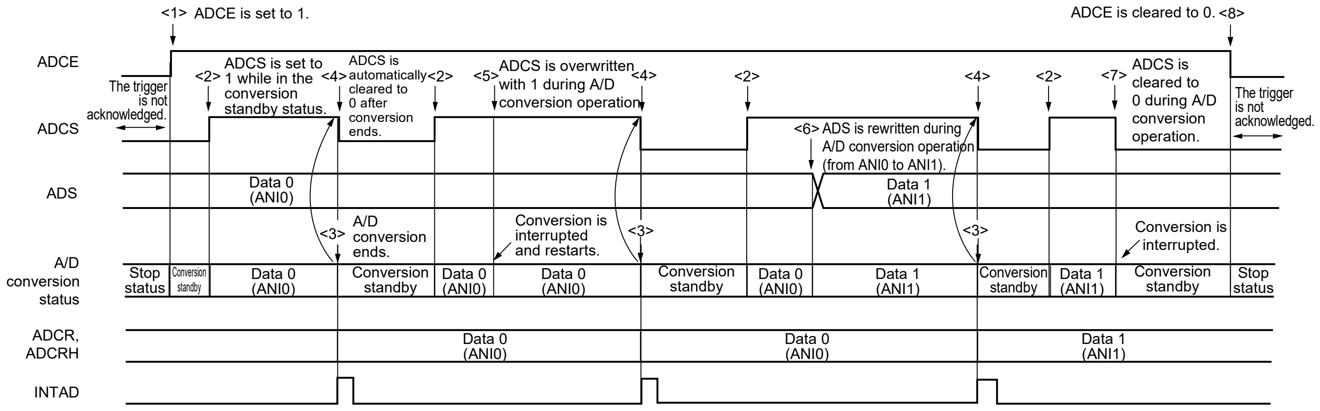
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 11-18. Example of Software Trigger Mode (One-Shot Conversion Mode) Operation Timing



11.6.3 Hardware trigger no-wait mode (sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1.0 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

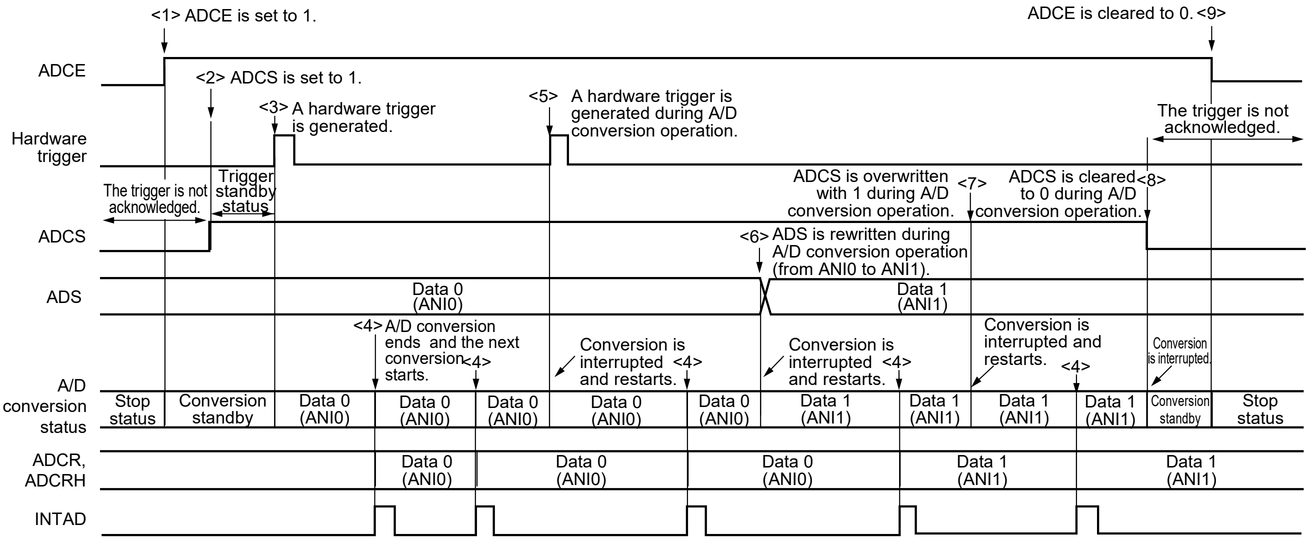
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-19. Example of Hardware Trigger No-Wait Mode (Sequential Conversion Mode) Operation Timing



11.6.4 Hardware trigger no-wait mode (one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1.0 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

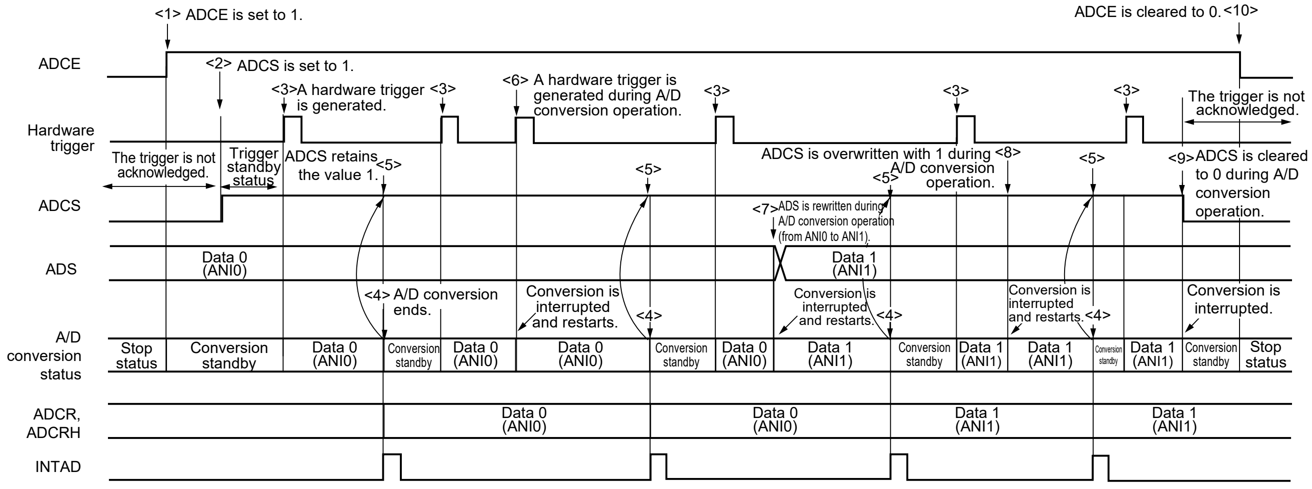
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-20. Example of Hardware Trigger No-Wait Mode (One-Shot Conversion Mode) Operation Timing



11.6.5 Hardware trigger wait 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.)

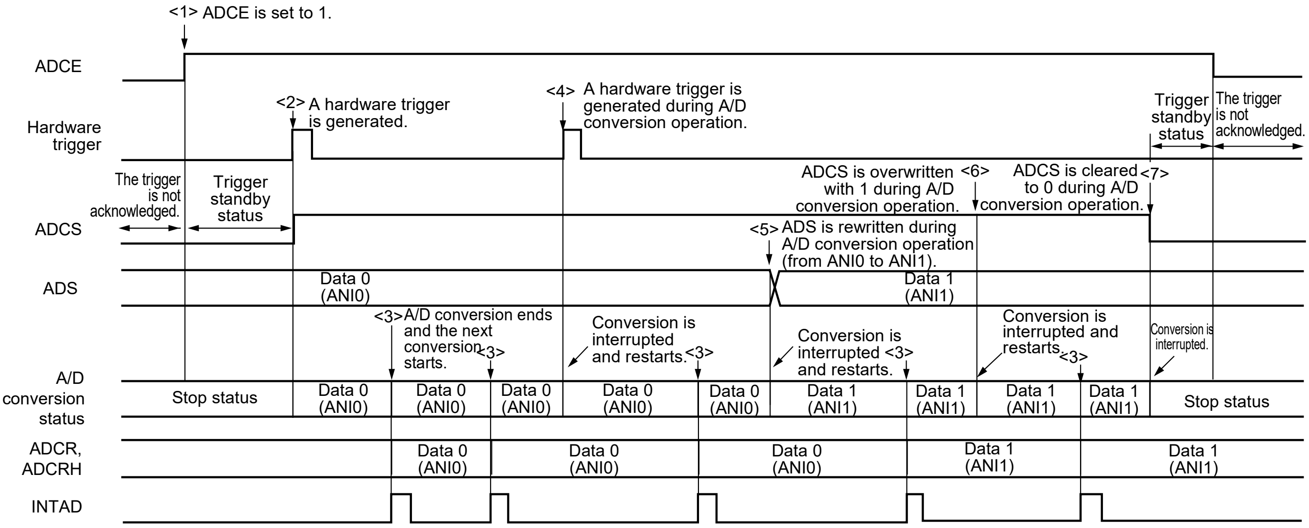
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 11-21. Example of Hardware Trigger Wait Mode (Sequential Conversion Mode) Operation Timing



11.6.6 Hardware trigger wait 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.

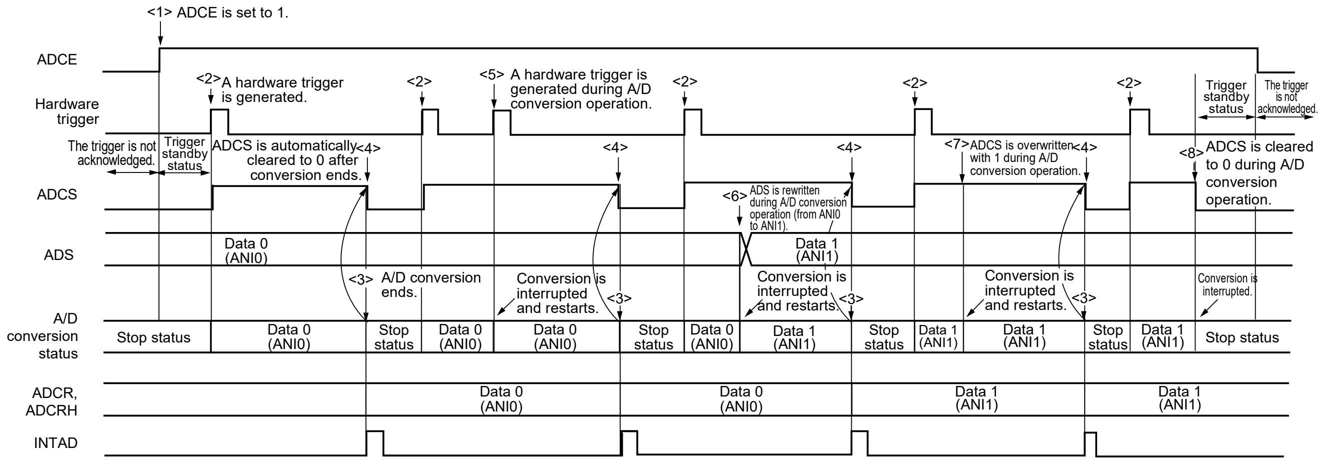
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 11-22. Example of Hardware Trigger Wait Mode (One-Shot Conversion Mode) Operation Timing

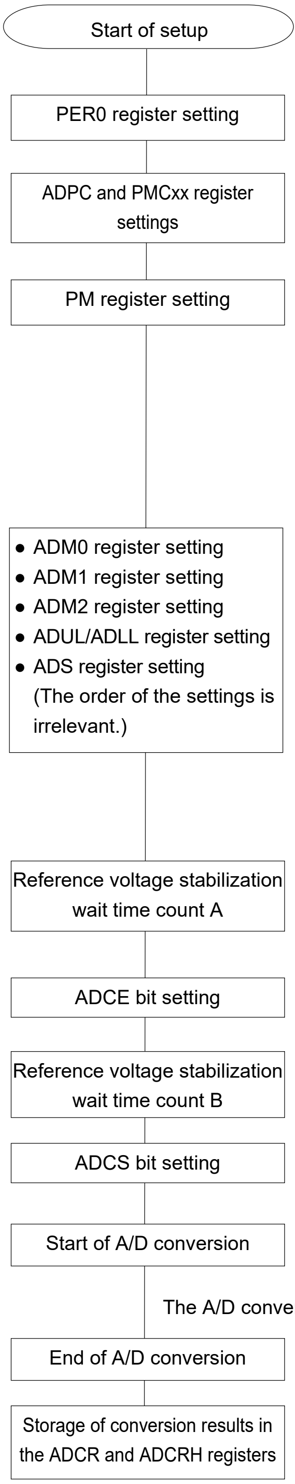


11.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

11.7.1 Setting up software trigger mode

Figure 11-23. Setting up Software Trigger Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 and ANI1 pins: Set using the ADPC register
ANI16 to ANI23 pins: Set using the PMCxx register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the software trigger mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADS4 to ADS0 bits: These are used to select the analog input channels.

The reference voltage stabilization wait time indicated by stabilization wait time count A is required when the value of the ADREFP1 and ADREFP0 bits is changed.
If change the ADREFP1 and ADREFP0 = 1, 0: A = 5 μs
If change the ADREFP1 and ADREFP0 = 0, 0 or 0, 1: No wait

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time B ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

The A/D conversion operations are performed.

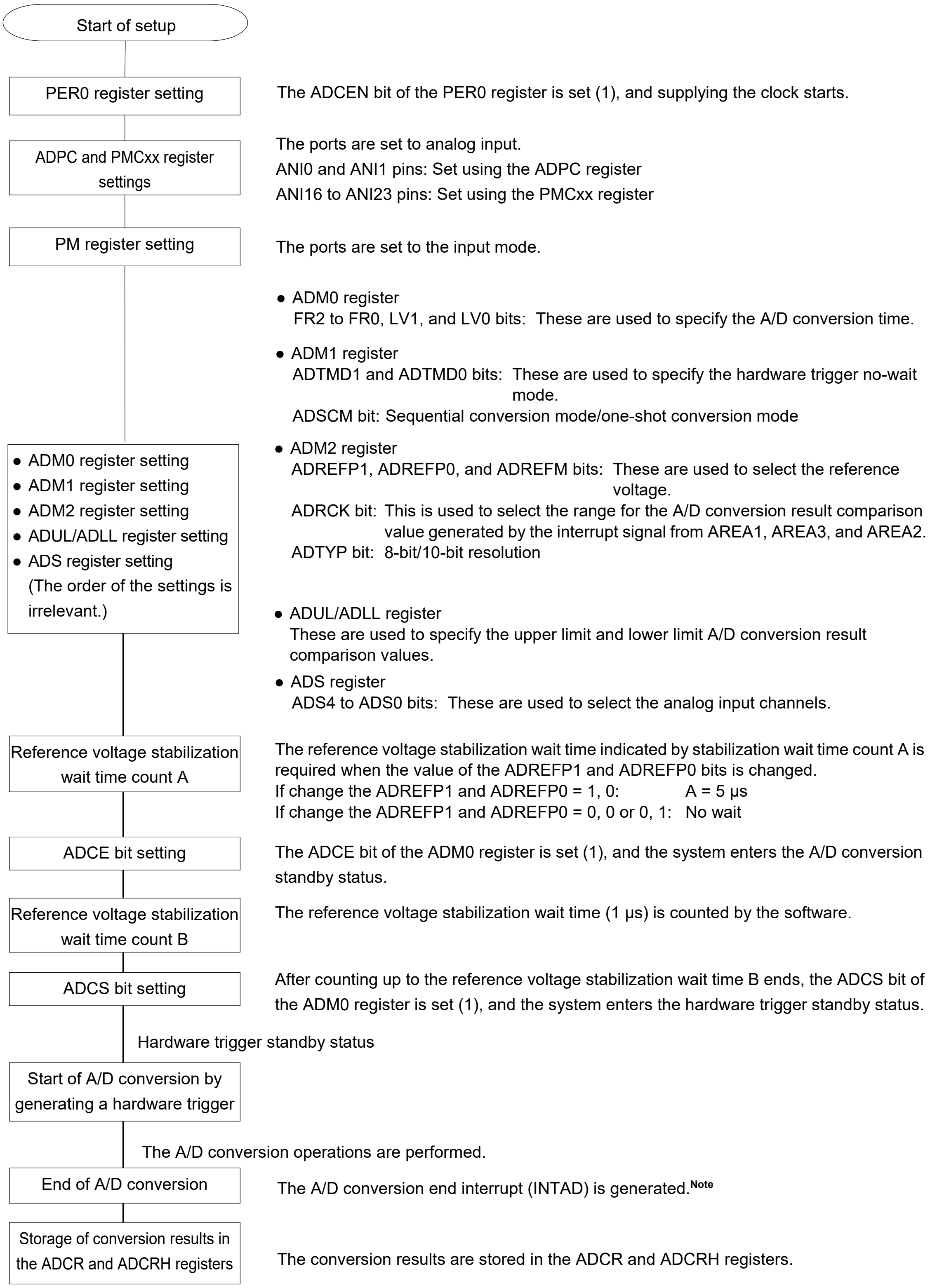
The A/D conversion end interrupt (INTAD) is generated.[Note]

The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

11.7.2 Setting up hardware trigger no-wait mode

Figure 11-24. Setting up Hardware Trigger No-Wait Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 and ANI1 pins: Set using the ADPC register
ANI16 to ANI23 pins: Set using the PMCxx register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the hardware trigger no-wait mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADS4 to ADS0 bits: These are used to select the analog input channels.

The reference voltage stabilization wait time indicated by stabilization wait time count A is required when the value of the ADREFP1 and ADREFP0 bits is changed.
If change the ADREFP1 and ADREFP0 = 1, 0: A = 5 μs
If change the ADREFP1 and ADREFP0 = 0, 0 or 0, 1: No wait

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time B ends, the ADCS bit of the ADM0 register is set (1), and the system enters the hardware trigger standby status.

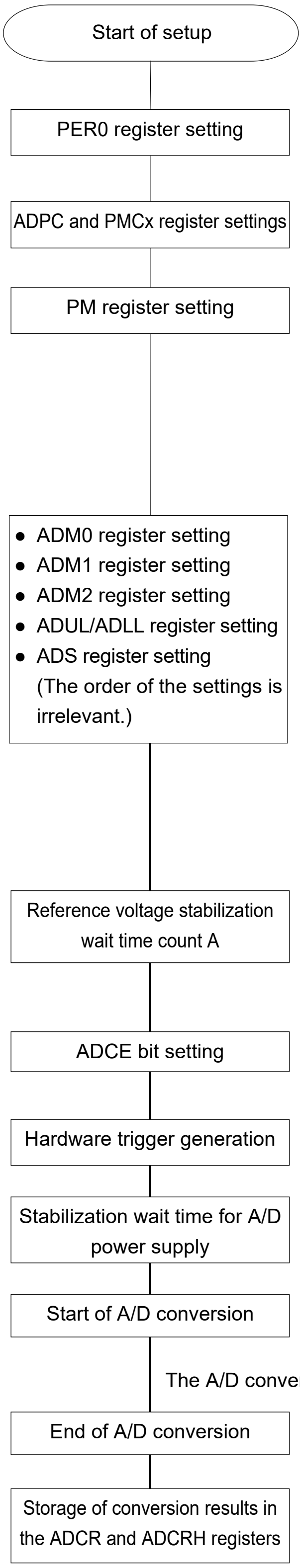
The A/D conversion end interrupt (INTAD) is generated.[Note]

The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

11.7.3 Setting up hardware trigger wait mode

Figure 11-25. Setting up Hardware Trigger Wait Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 and ANI1 pins: Set using the ADPC register
ANI16 to ANI23 pins: Set using the PMCx register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the hardware trigger wait mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
  ADTRS1 and ADTRS0 bits: These are used to select the hardware trigger signal.
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADS4 to ADS0 bits: These are used to select the analog input channels.

The reference voltage stabilization wait time indicated by stabilization wait time count A is required when the value of the ADREFP1 and ADREFP0 bits is changed.
If change the ADREFP1 and ADREFP0 = 1, 0: A = 5 μs
If change the ADREFP1 and ADREFP0 = 0, 0 or 0, 1: No wait

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The system automatically counts up to the stabilization wait time for A/D power supply.

After counting up to the reference voltage stabilization wait time ends, A/D conversion starts

The A/D conversion operations are performed.

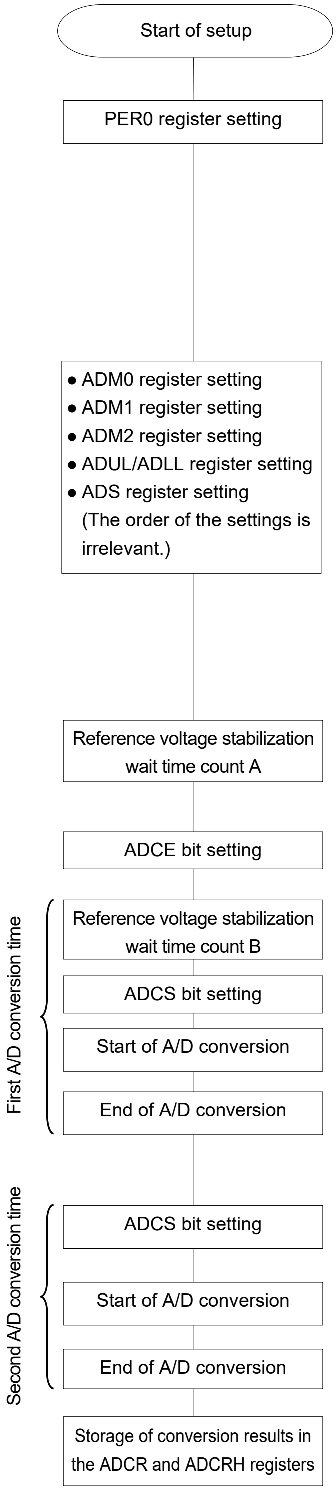
The A/D conversion end interrupt (INTAD) is generated.[Note]

The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

11.7.4 Setup when temperature sensor output/internal reference voltage output is selected (example for software trigger mode and one-shot conversion mode)

Figure 11-26. Setup when temperature sensor output voltage/internal reference voltage is selected



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the software trigger mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADISS and ADS4 to ADS0 bits: These are used to select the temperature sensor output voltage or internal reference voltage.

The reference voltage stabilization wait time indicated by stabilization wait time count A is required when the value of the ADREFP1 and ADREFP0 bits is changed.
If change the ADREFP1 and ADREFP0 = 0, 0 or 0, 1: No wait
If change the ADREFP1 and ADREFP0 = 1, 0: Setting prohibited

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time B ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts

The A/D conversion end interrupt (INTAD) will be generated.
After ADISS is set (1), the initial conversion result cannot be used.

The ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

The A/D conversion end interrupt (INTAD) is generated. Note

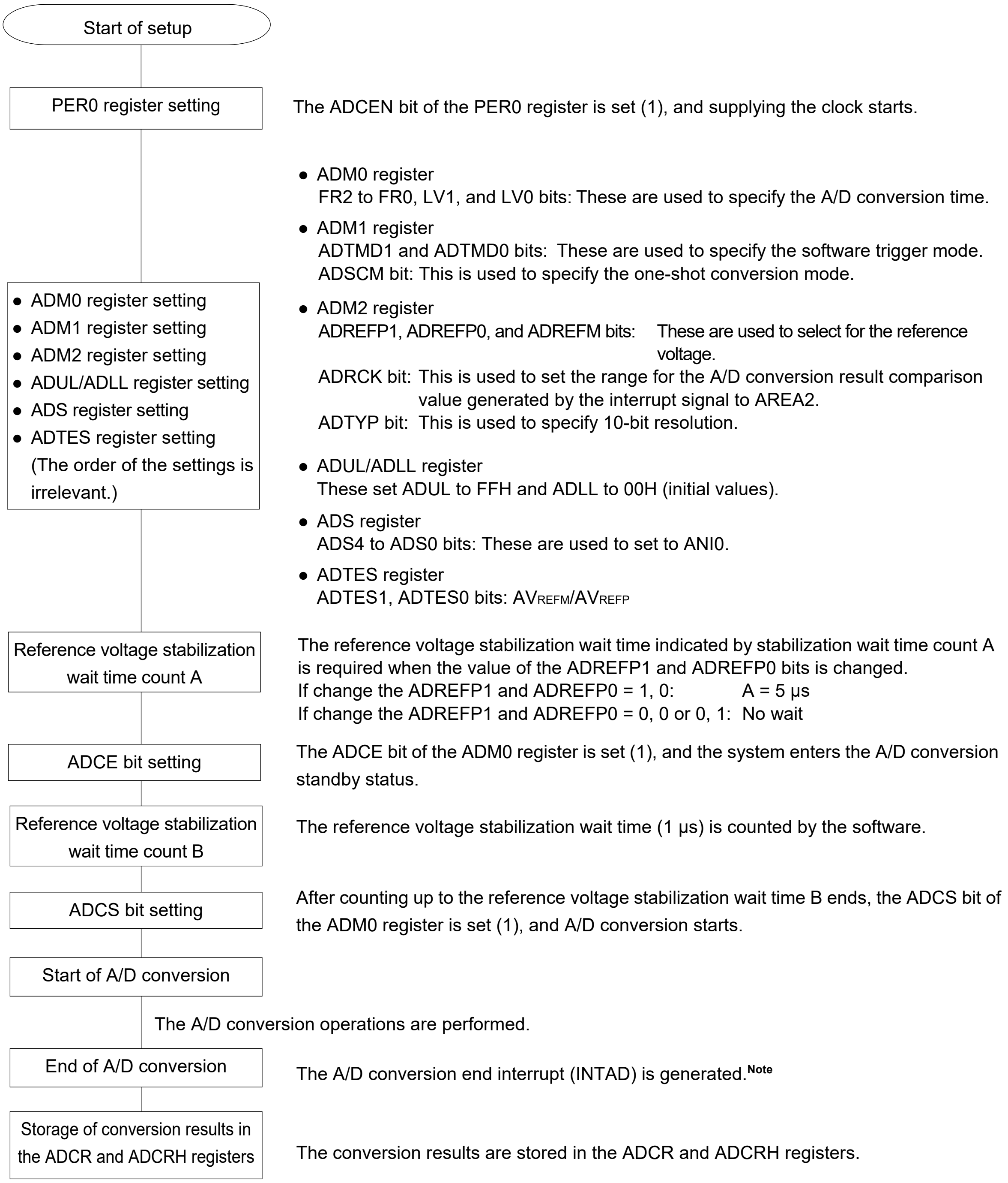
The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

Caution This setting can be used only in HS (high-speed main) mode.

11.7.5 Setting up test mode

Figure 11-27. Setting up Test Trigger Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the software trigger mode.
  ADSCM bit: This is used to specify the one-shot conversion mode.
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select for the reference voltage.
  ADRCK bit: This is used to set the range for the A/D conversion result comparison value generated by the interrupt signal to AREA2.
  ADTYP bit: This is used to specify 10-bit resolution.
- ADUL/ADLL register
  These set ADUL to FFH and ADLL to 00H (initial values).
- ADS register
  ADS4 to ADS0 bits: These are used to set to ANI0.
- ADTES register
  ADTES1, ADTES0 bits: AV_{REFM}/AV_{REFP}

The reference voltage stabilization wait time indicated by stabilization wait time count A is required when the value of the ADREFP1 and ADREFP0 bits is changed.
If change the ADREFP1 and ADREFP0 = 1, 0: A = 5 μs
If change the ADREFP1 and ADREFP0 = 0, 0 or 0, 1: No wait

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time B ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

The A/D conversion operations are performed.

The A/D conversion end interrupt (INTAD) is generated.^{Note}

The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

Caution For the procedure for testing the A/D converter, see 20.3.8 A/D test function.

11.8 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.

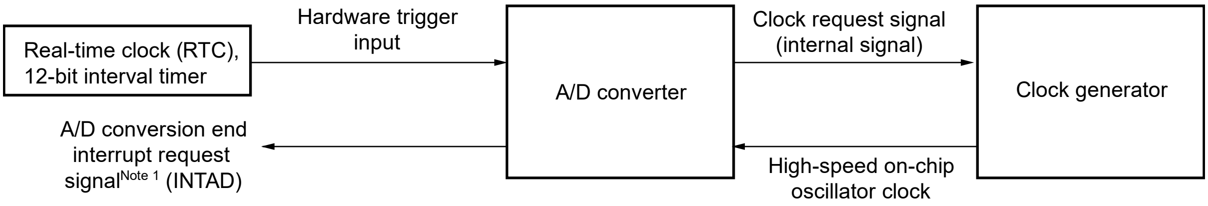
If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time in SNOOZE mode. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

In the SNOOZE mode, only the following two conversion modes can be used:

- Hardware trigger wait mode (one-shot conversion mode)

Caution That the SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

Figure 11-28. 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 **11.7.3 Setting up hardware trigger wait mode[Note 2]**.) Just before move to STOP mode, bit 2 (AWC) of A/D converter mode register 2 (ADM2) is set to 1. After the initial settings are specified, bit 0 (ADCE) of A/D converter mode register 0 (ADM0) is set to 1.

If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the system automatically counts up to the A/D power supply stabilization wait time, and then A/D conversion starts.

The SNOOZE mode operation after A/D conversion ends differs depending on whether an interrupt signal is generated[Note 1].

Notes 1. Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL/ADLL register), there is a possibility of no interrupt signal being generated.

2. Be sure to set the ADM1 register to E2H or E3H.

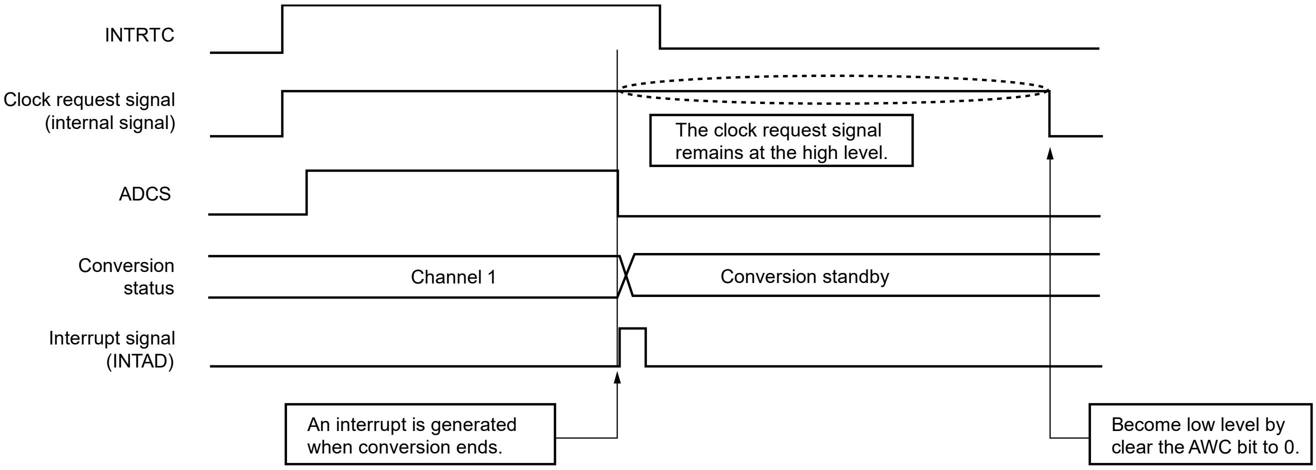
Remark The hardware trigger is INTRTC and INTIT.
Specify the hardware trigger by using the A/D Converter Mode Register 1 (ADM1).

(1) If an interrupt is generated after A/D conversion ends

If the A/D conversion result value is inside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is generated.

When A/D conversion ends and an A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter returns to normal operation mode from SNOOZE mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of the A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

Figure 11-29. Operation Example When Interrupt Is Generated After A/D Conversion Ends

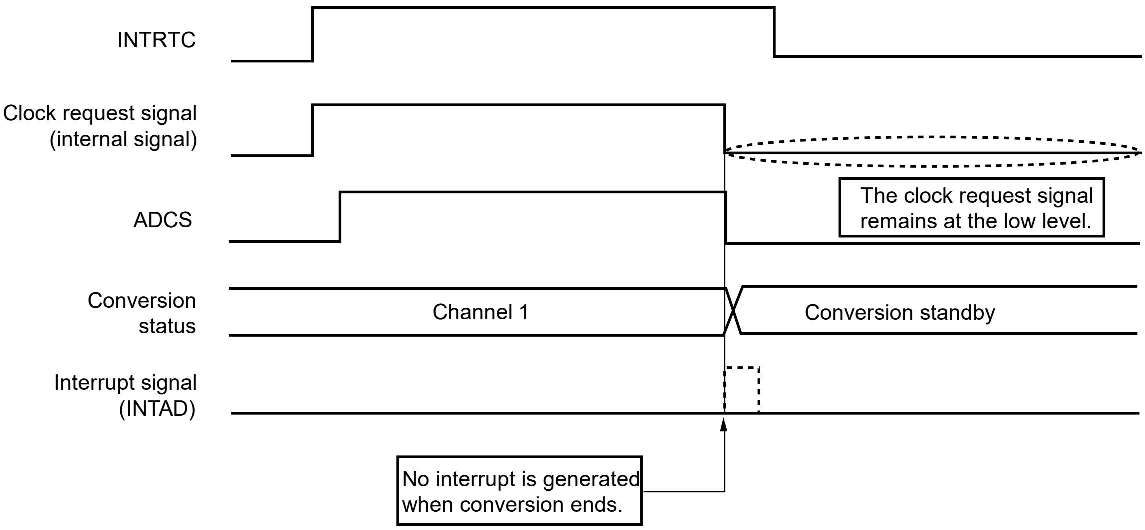


(2) If no interrupt is generated after A/D conversion ends

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.

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.

Figure 11-30. Operation Example When No Interrupt Is Generated After A/D Conversion Ends



(3) Operation when A/D conversion is interrupted or resumed

If A/D conversion is interrupted (by clearing bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0), the clock request signal (an internal signal) is set to the low level, and supplying the high-speed on-chip oscillator clock stops. When another hardware trigger is input, the clock request signal is set to the high level, supplying the high-speed on-chip oscillator clock resumes, and A/D conversion starts in the SNOOZE mode.

Figure 11-31. Example of Operation When A/D Conversion Is Interrupted or Resumed

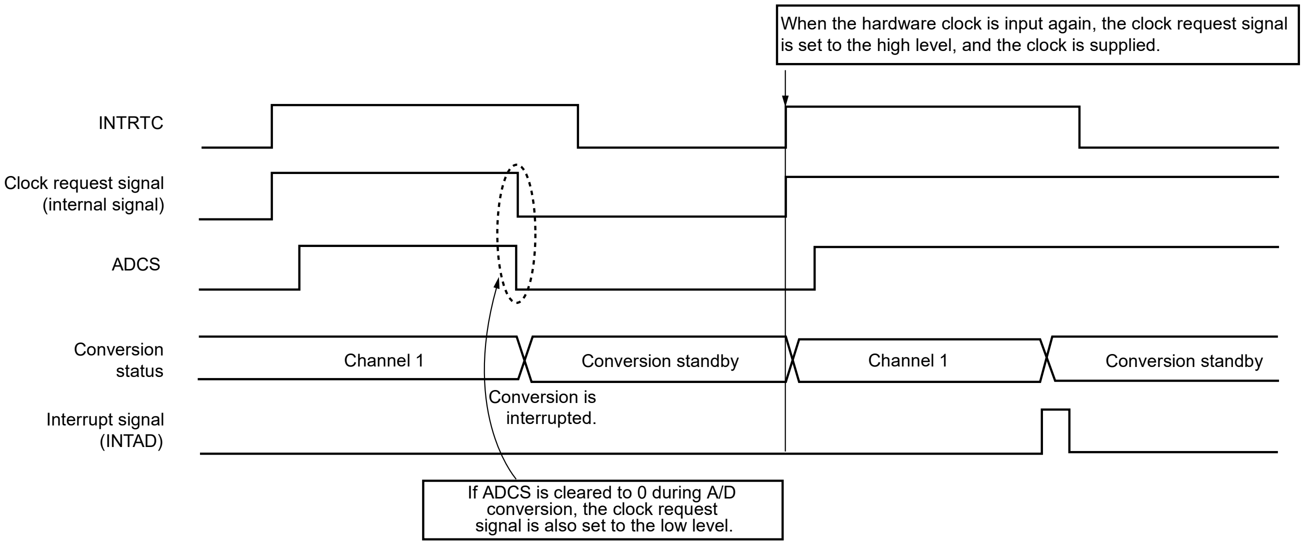
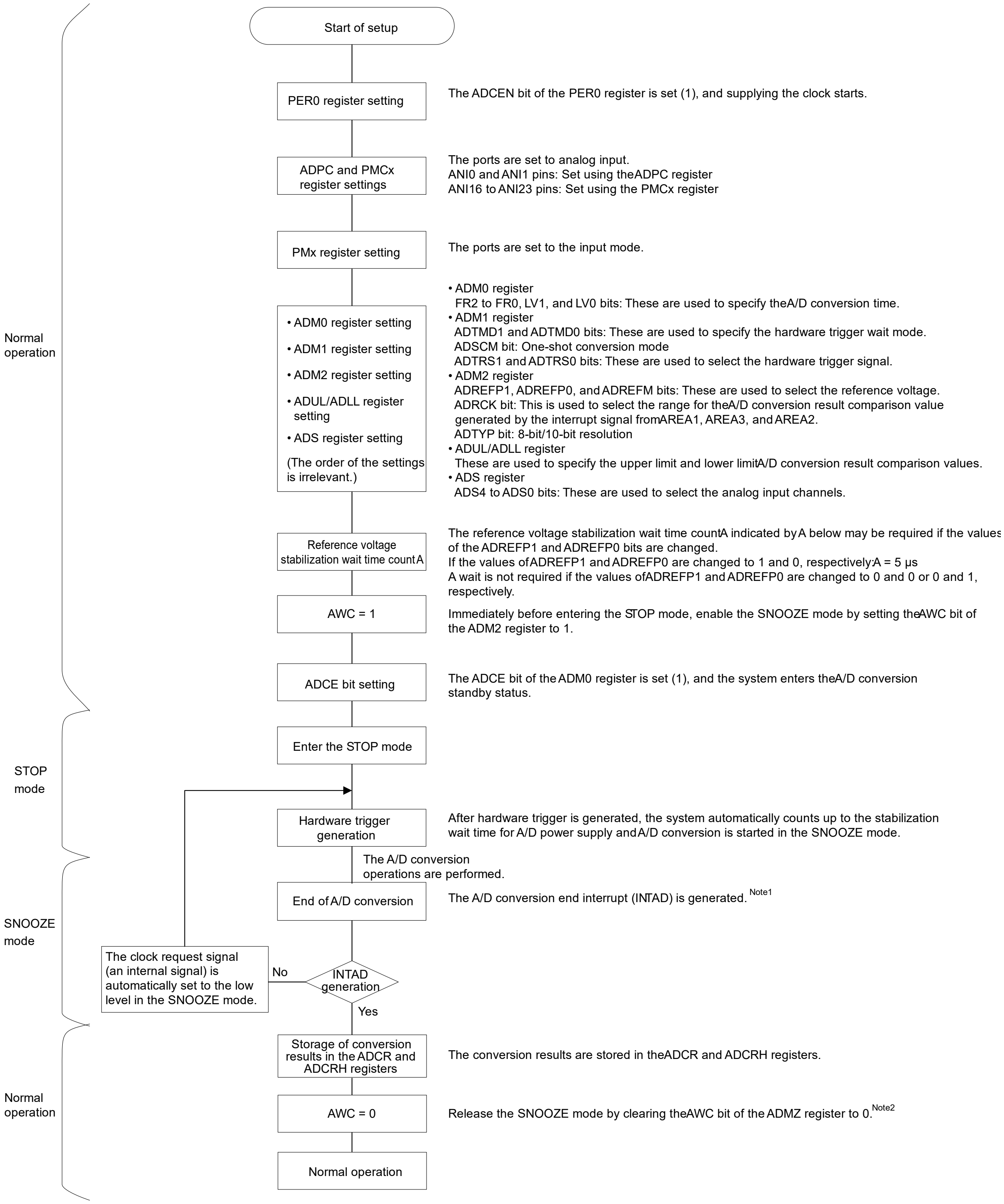


Figure 11-32. Flowchart for Setting up SNOOZE Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 and ANI1 pins: Set using the ADPC register
ANI16 to ANI23 pins: Set using the PMCx register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the hardware trigger wait mode.
  ADSCM bit: One-shot conversion mode
  ADTRS1 and ADTRS0 bits: These are used to select the hardware trigger signal.
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADS4 to ADS0 bits: These are used to select the analog input channels.

The reference voltage stabilization wait time count A indicated by A below may be required if the values of the ADREFP1 and ADREFP0 bits are changed.
If the values of ADREFP1 and ADREFP0 are changed to 1 and 0, respectively: A = 5 μs
A wait is not required if the values of ADREFP1 and ADREFP0 are changed to 0 and 0 or 0 and 1, respectively.

Immediately before entering the STOP mode, enable the SNOOZE mode by setting the AWC bit of the ADM2 register to 1.

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

After hardware trigger is generated, the system automatically counts up to the stabilization wait time for A/D power supply and A/D conversion is started in the SNOOZE mode.

The A/D conversion end interrupt (INTAD) is generated. Note1

The conversion results are stored in the ADCR and ADCRH registers.

Release the SNOOZE mode by clearing the AWC bit of the ADMZ register to 0. Note2

Notes 1. If the A/D conversion end interrupt request signal (INTAD) is not generated by setting ADRCK bit and ADUL/ADLL register, the result is not stored in the ADCR and ADCRH registers.
The system enters the STOP mode again. If a hardware trigger is input later, A/D conversion operation is again performed in the SNOOZE mode.

2. If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE or normal operation mode. Be sure to clear the AWC bit to 0.

11.9 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.

$$1\text{LSB} = 1/2^{10} = 1/1024$$
$$= 0.098\%\text{FSR}$$

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 11-33. Overall Error

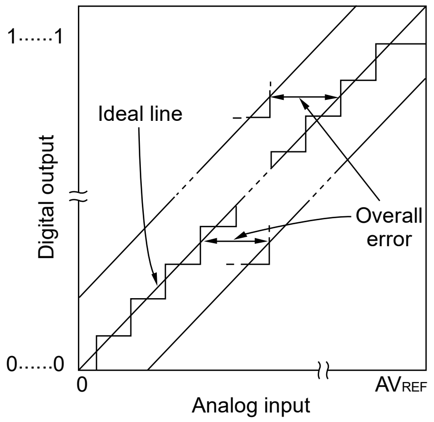
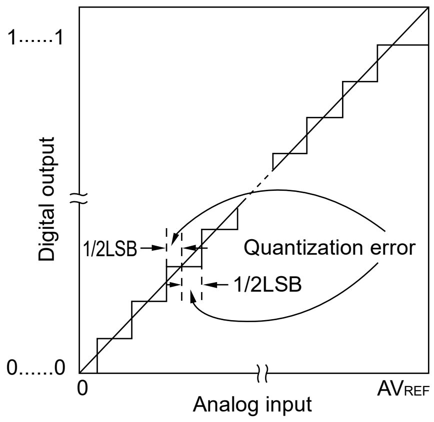


Figure 11-34. Quantization Error



(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 11-35. Zero-Scale Error

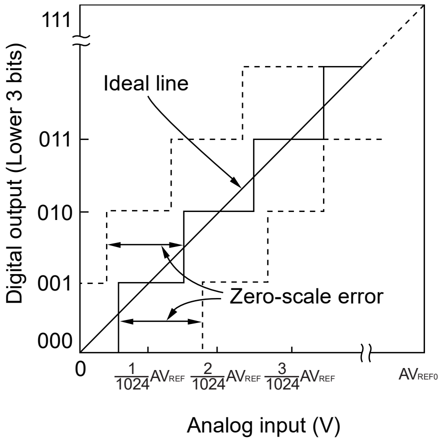


Figure 11-36. Full-Scale Error

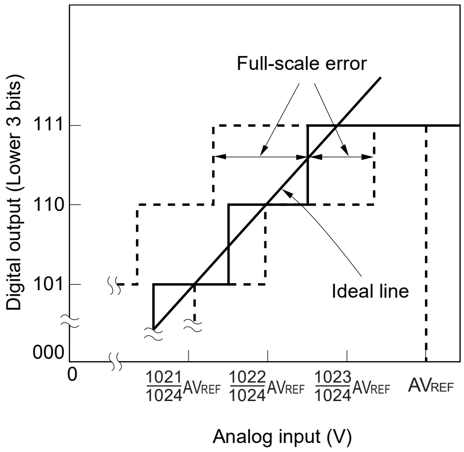


Figure 11-37. Integral Linearity Error

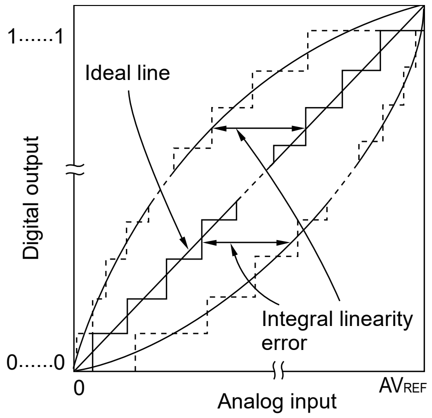
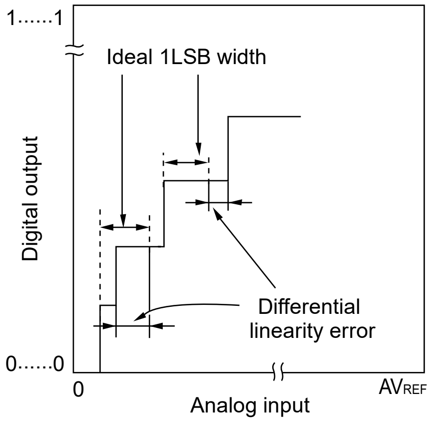


Figure 11-38. Differential Linearity Error



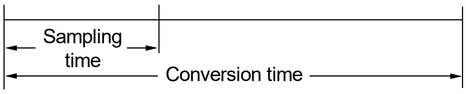
(8) Conversion time

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.



11.10 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, ANI1 and ANI16 to ANI23 pins

Observe the rated range of the ANI0, ANI1 and ANI16 to ANI23 pins input voltage. If a voltage exceeding V_{DD} and AV_{REFP} or below V_{SS} and AV_{REFM} (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

When internal reference voltage (1.45 V) is selected as the reference voltage source for the + side of the A/D converter, do not input voltage exceeding internal reference voltage (1.45 V) to a pin selected by the ADS register. However, the input of voltages exceeding the internal reference voltage (1.45 V) to pins not selected by the ADS register does not create a problem.

Caution Internal reference voltage (1.45 V) can be used only in HS (high-speed main) mode.

(3) Conflicting operations

<1> Conflict between the A/D conversion result register (ADCR, ADCRH) write and the ADCR or ADCRH register read by instruction upon the end of conversion
The ADCR or ADCRH register read has priority. After the read operation, the new conversion result is written to the ADCR or ADCRH registers.

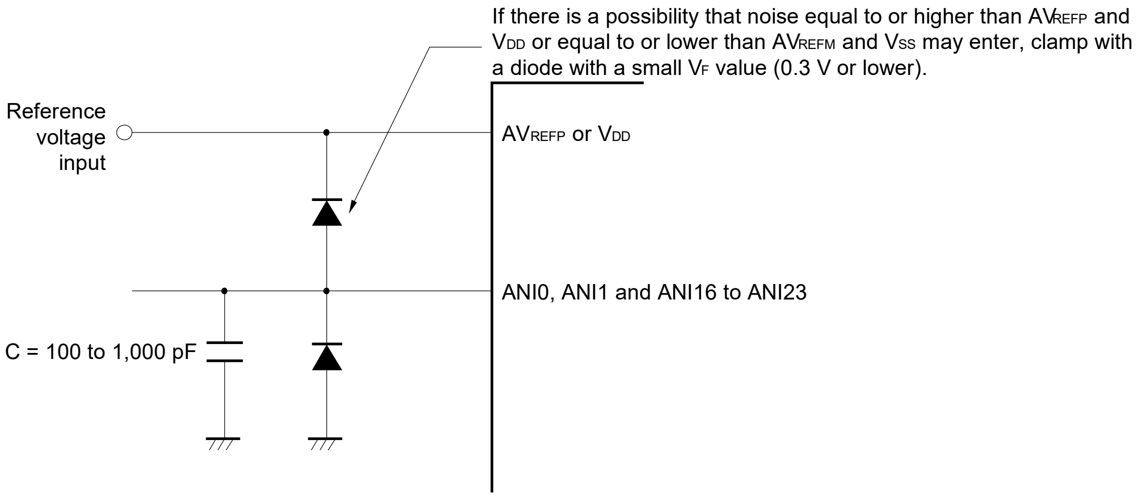
<2> Conflict between the ADCR or ADCRH register write and the A/D converter mode register 0 (ADM0) write, the analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion
The ADM0, ADS, or ADPC registers write has priority. The ADCR or ADCRH register write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REFP}, V_{DD}, ANI0, ANI1 and ANI16 to ANI23 pins.

<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.
<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting an external capacitor as shown in **Figure 11-39** is recommended.
<3> Do not switch these pins with other pins during conversion.
<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 11-39. Analog Input Pin Connection



(5) Analog input (ANIn) pins

<1> The analog input pins (ANI0 and ANI1) are also used as input port pins (P20 and P21). When A/D conversion is performed with any of the ANI0 and ANI1 pins selected, do not change to output value P20 and P21 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to avoid the input or output of digital signals and signals with similarly sharp transitions during conversion.

(6) Input impedance of analog input (ANIn) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.

Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.

To make sure that sampling is effective, however, we recommend using the converter with analog input sources that have output impedances no greater than 1 kΩ. If a source has a higher output impedance, lengthen the sampling time or connect a larger capacitor (with a value of about 0.1 μF) to the pin from among ANI0, ANI1, and ANI16 to ANI23 to which the source is connected (see **Figure 11-39**). The sampling capacitor may be being charged while the setting of the ADCS bit is 0 and immediately after sampling is restarted and so is not defined at these times. Accordingly, the state of conversion is undefined after charging starts in the next round of conversion after the value of the ADCS bit has been 1 or when conversion is repeated. Thus, to secure full charging regardless of the size of fluctuations in the analog signal, ensure that the output impedances of the sources of analog inputs are low or secure sufficient time for the completion of conversion.

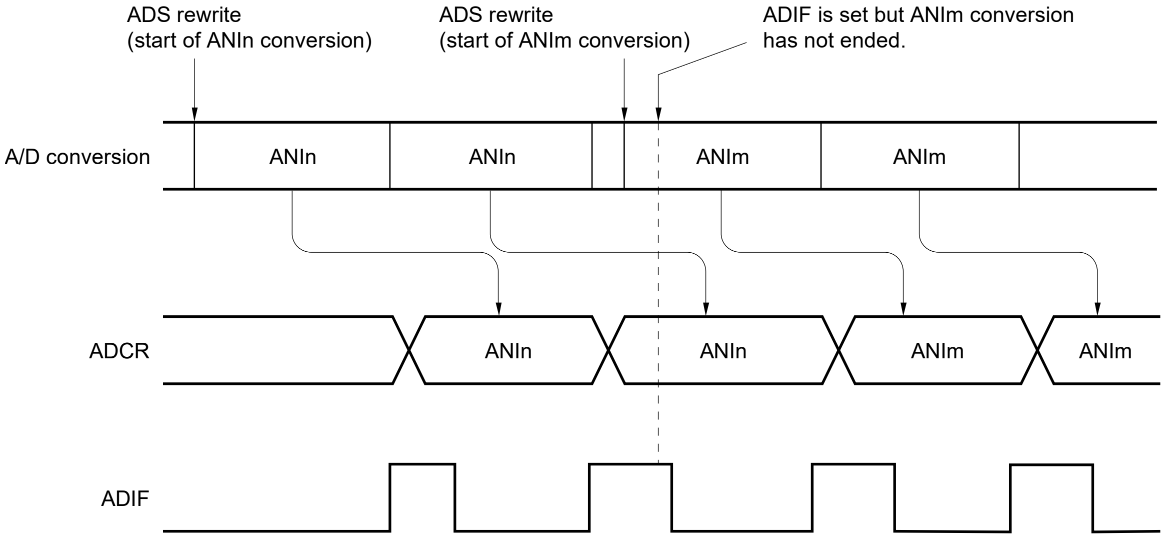
(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF flag for the pre-change analog input may be set just before the ADS register rewrite. Caution is therefore required since, at this time, when ADIF flag is read immediately after the ADS register rewrite, ADIF flag is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

Figure 11-40. Timing of A/D Conversion End Interrupt Request Generation



(8) Conversion results just after A/D conversion start

In software trigger mode and hardware trigger no-wait mode, if the ADCE bit is set to 1 and then the ADCS bit is set to 1 before 1.0 μs elapses, the A/D conversion value immediately after A/D conversion starts might not satisfy the ratings. In this case, take measures such as polling the A/D conversion end interrupt request signal (INTAD) and discarding the first conversion result.

(9) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), A/D port configuration register (ADPC), and port mode control register (PMC), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, ADPC, or PMC register. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 11-41. Internal Equivalent Circuit of ANIn Pin

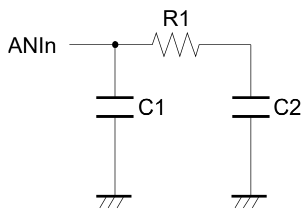


Table 11-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REFP}, V_{DD}	ANIn Pins	R1 [kΩ]	C1 [pF]	C2 [pF]
3.6 V ≤ V_{DD} ≤ 5.5 V	ANI0 and ANI1	14	8	2.5
	ANI16 to ANI23	18	8	7.0
2.7 V ≤ V_{DD} ≤ 3.6 V	ANI0 and ANI1	39	8	2.5
	ANI16 to ANI23	53	8	7.0
1.8 V ≤ V_{DD} ≤ 2.7 V	ANI0 and ANI1	231	8	2.5
	ANI16 to ANI23	321	8	7.0
1.6 V ≤ V_{DD} < 2.7 V	ANI0 and ANI1	632	8	2.5
	ANI16 to ANI23	902	8	7.0

Caution The A/D converter’s internal voltage cannot be used in SNOOZE mode.

Remark The resistance and capacitance values shown in Table 11-4 are not guaranteed values.

(11) Starting the A/D converter

Start the A/D converter after the AV_{REFP} and V_{DD} voltages stabilize.

CHAPTER 12 SERIAL ARRAY UNIT

Serial array unit has two serial channels. Each channel can achieve Simplified SPI (CSI [Note]), and UART.
Function assignment of each channel supported by the R7F0C001G/L, R7F0C002G/L is as shown below.

<R> **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.

Channel	Used as Simplified SPI (CSI)	Used as UART
0	CSI00	UART0
1	CSI01	

12.1 Functions of Serial Array Unit

Each serial interface supported by the R7F0C001G/L, R7F0C002G/L has the following features.

12.1.1 Simplified SPI (CSI00, CSI01)

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 **12.5 Operation of Simplified SPI (CSI00, CSI01) 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
  During master communication (CSI00): Max. $f_{MCK}/2$ [Notes 1, 2]
  During master communication (CSI01): Max. $f_{MCK}/4$ [Note 2]
  During slave communication: Max. $f_{MCK}/6$ [Note 2]

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

Notes 1. In master communication (CSI00), maximum transfer rate become $f_{MCK}/2$ when the following conditions.

- $2.7\ V \le EV_{DD} = V_{DD} \le 5.5\ V$
- $f_{MCK} \le 12$ MHz

Other cases, maximum transfer rate become $f_{MCK}/4$.

2. Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics (see **CHAPTER 27 ELECTRICAL SPECIFICATIONS**).

12.1.2 UART (UART0)

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).

For details about the settings, see **12.6 Operation of UART (UART0) Communication.**

[Data transmission/reception]

- Data length of 7, 8, or 9 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

In addition, UART0 supports the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

12.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 12-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	9 bits
Buffer register	Lower 9 bits of serial data register mn (SDRmn)[Note]
Serial clock I/O	SCK00, SCK01 pins (for Simplified SPI)
Serial data input	SI00, SI01 pins (for Simplified SPI), R_xD0 pin (for UART)
Serial data output	SO00, SO01 pins (for Simplified SPI), T_xD0 pin (for UART), output controller
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) ● 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 register 1 (PIM1) ● Port output mode register 1 (POM1) ● LCD port function registers 0, 3 (PFSEG0, PFSEG3) ● Port mode register 1 (PM1) ● Port register 1 (P1)

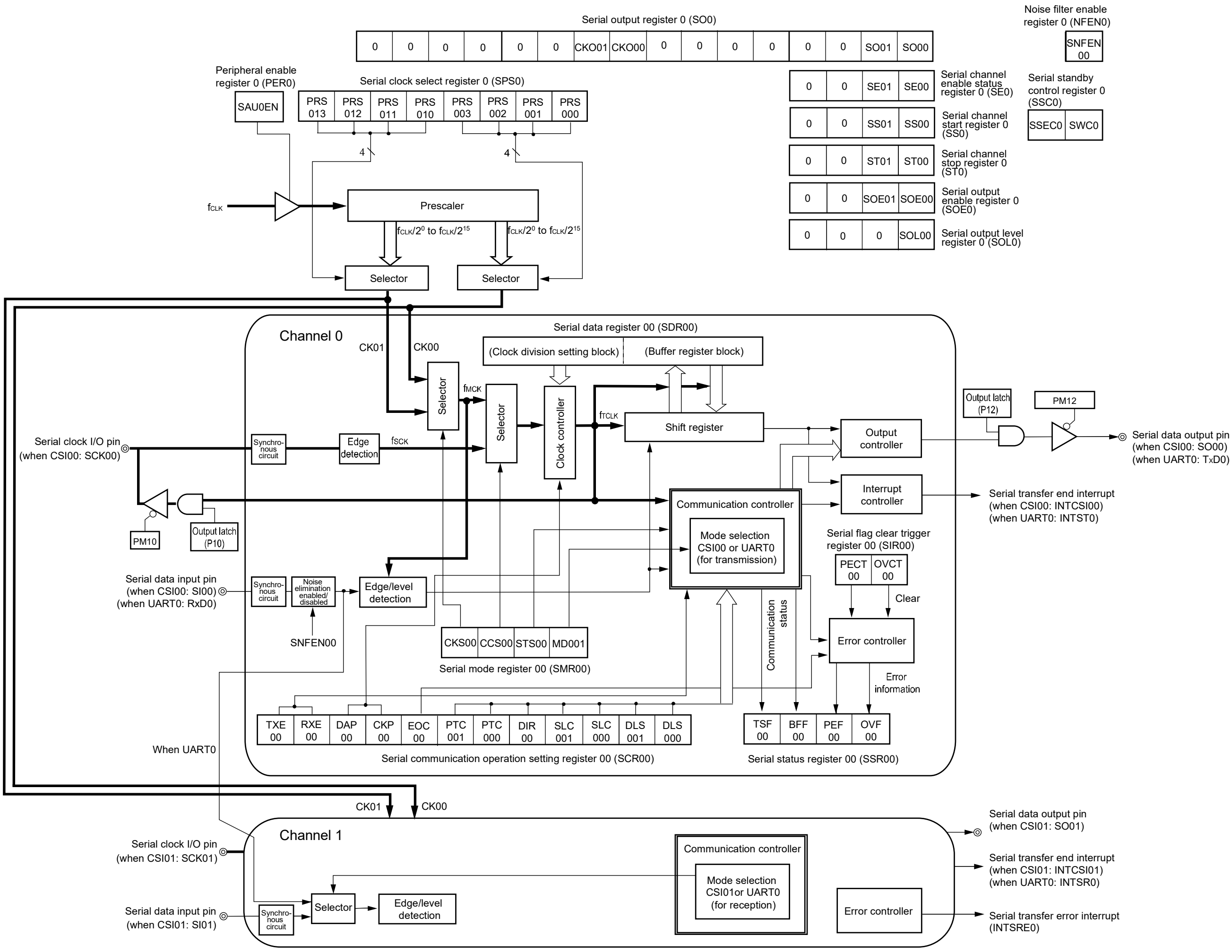
Note The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

- CSIp communication … SIOp (CSIp data register)
- UARTq reception … RXDq (UARTq receive data register)
- UARTq transmission … TXDq (UARTq transmit data register)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01),
q: UART number (q = 0)

Figure 12-1 shows the block diagram of the serial array unit.

Figure 12-1. Block Diagram of Serial Array Unit



12.2.1 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.

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 9 bits of serial data register mn (SDRmn).

	8	7	6	5	4	3	2	1	0
Shift register									

12.2.2 Lower 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) 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 9 bits. When data is to be transmitted, set transmit to be transferred to the shift register to the lower 9 bits.

The data stored in the lower 9 bits of this register is as follows, depending on the setting of bits 0 and 1 (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) (settable in UART0 mode only)

The SDRmn register can be read or written in 16-bit units.

The lower 8 bits of the SDRmn register can be read or written [Note] as the following SFR, depending on the communication mode.

- CSIp communication … SIOp (CSIp data register)
- UARTq reception … RXDq (UARTq receive data register)
- UARTq transmission … TXDq (UARTq transmit data register)

Reset signal generation clears the SDRmn register to 0000H.

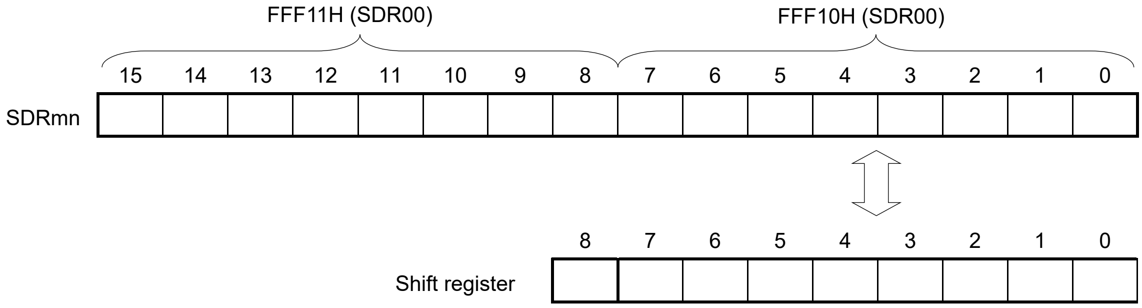
Note When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remarks 1. After data is received, “0” is stored in bits 0 to 8 in bit portions that exceed the data length.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01),
q: UART number (q = 0)

Figure 12-2. Format of Serial Data Register mn (SDRmn) (mn = 00, 01)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W



Remark For the function of the higher 7 bits of the SDRmn register, see **12.3 Registers Controlling Serial Array Unit**.

12.3 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)
- Noise filter enable register 0 (NFEN0)
- Port input mode register 1 (PIM1)
- Port output mode register 1 (POM1)
- LCD port function registers 0, 3 (PFSEG0, PFSEG3)
- Port mode register 1 (PM1)
- Port registers 1 (P1)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.3.1 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 is used, be sure to set bit 2 (SAU0EN) of this register to 1.

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

Reset signal generation clears the PER0 register to 00H.

Figure 12-3. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

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

Cautions 1. When setting serial array unit, be sure to first set the following registers with the SAU0EN bit set to 1. If SAU0EN = 0, control registers of serial array unit m become default values and writing to them is ignored (except for the noise filter enable register 0 (NFEN0), port input mode register 1 (PIM1), port output mode register 1 (POM1), LCD port function registers 0, 3 (PFSEG0, PFSEG3), port mode register 1 (PM1), and port register 1 (P1)).

- **Serial clock select register m (SPSm)**
- **Serial mode register mn (SMRmn)**
- **Serial communication operation setting register mn (SCRmn)**
- **Serial data register mn (SDRmn)**
- **Serial flag clear trigger register mn (SIRmn)**
- **Serial status register mn (SSRmn)**
- **Serial channel start register m (SSm)**
- **Serial channel stop register m (STm)**
- **Serial channel enable status register m (SEm)**
- **Serial output enable register m (SOEm)**
- **Serial output level register m (SOLm)**
- **Serial output register m (SOm)**
- **Serial standby control register m (SSCm)**

2. Be sure to clear bits 1, 3, 4, 6 to “0”.

12.3.2 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 12-4. Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Section of operation clock (CKmk) [Note]					
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 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 kHz	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".

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency
f_{SUB}: Subsystem clock frequency

2. m: Unit number (m = 0)

3. k = 0, 1

12.3.3 Serial mode register mn (SMRmn)

The SMRmn register is a register that sets an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, an operation mode (Simplified SPI (CSI), or UART), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMRmn register is prohibited when the register is in operation (when SEmn = 1). However, the MDmn0 bit can be rewritten during operation.

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

Reset signal generation sets the SMRmn register to 0020H.

Figure 12-5. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00), F0112H, F0113H (SMR01) 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$^{\text{Note}}$	0	SIS mn0 Note	1	0	0	0	MD mn1	MD mn0

CKS mn	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.	

CCS mn	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.	

STS mn	Selection of start trigger source
0	Only software trigger is valid (selected for Simplified SPI (CSI) and UART transmission).
1	Valid edge of the R_XDq 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 register only.

Caution Be sure to clear bits 13 to 6, and 4 to 2 for the SMR00 register, or bits 13 to 9, 7, 4 to 2 for the SMR01 register to "0". And be sure to set bit 5 to "1".

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01),
q: UART number (q = 0)

Figure 12-5. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00), F0112H, F0113H (SMR01) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn Note	0	SIS mn0 Note	1	0	0	0	MD mn1	MD mn0

SIS mn0	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MD mn1	Setting of operation mode of channel n
0	Simplified SPI (CSI) mode
1	UART mode

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting the MDmn0 bit to 1 when SDRmn data has run out.	

Note The SMR01 register only.

Caution Be sure to clear bits 13 to 6, and 4 to 2 for the SMR00 register, or bits 13 to 9, 7, 4 to 2 for the SMR01 register to "0". And be sure to set bit 5 to "1".

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01),
q: UART number (q = 0)

12.3.4 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.

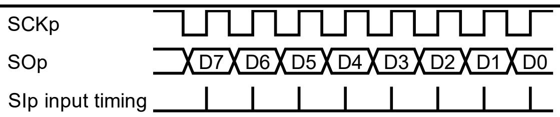
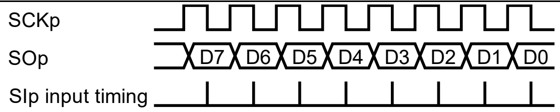
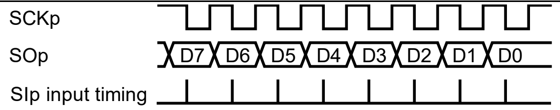
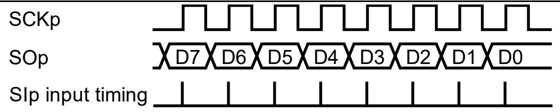
Reset signal generation sets the SCRmn register to 0087H.

Figure 12-6. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00), F011AH, F011BH (SCR01) 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	SLC mn1 Note 1	SLC mn0	0	1	DLS mn1	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.			

EOC mn	Selection of masking of error interrupt signal (INTSREx (x = 0 to 3))
0	Masks error interrupt INTSREx (INTSRx is not masked).
1	Enables generation of error interrupt INTSREx (INTSRx is masked if an error occurs).
Set EOCmn = 0 in the Simplified SPI (CSI) mode, and during UART transmission Note 2.	

Notes **1.** The SCR00 register only.

2. When using CSIp not with EOCmn = 0, error interrupt INTSREn may be generated.

Caution **Be sure to clear the following bits to “0”.**
SCR00: bits 11, 6, 3
SCR01: bits 11, 6, 5, 3
Be sure to set bit 2 to “1”.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-6. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00), F011AH, F011BH (SCR01) 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	SLC mn1 Note 1	SLC mn0	0	1	DLS mn1	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 2.	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.			

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.

SLC mn1 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. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the Simplified SPI (CSI) mode.		

DLS mn1	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 UART0 mode only)
1	0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	1	8-bit data length (stored in bits 0 to 7 of the SDRmn register)
Other than above		Setting prohibited

Notes 1. The SCR00 register only.

2. 0 is always added regardless of the data contents.

Caution Be sure to clear the following bits to "0".
SCR00: bits 11, 6, 3
SCR01: bits 11, 6, 5, 3
Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

12.3.5 Higher 7 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) function as a transmit/receive buffer register, and bits 15 to 9 (higher 7 bits) are used as a register that sets the division ratio of the operation clock (f_{MCK}).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by bits 15 to 9 (higher 7 bits) of the SDRmn register is used as the transfer clock.

The lower 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 9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 9 bits.

If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (higher 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 be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 9 bits of the SDRmn register. When the SDRmn register is read during operation, 0 is always read.

Reset signal generation clears the SDRmn register to 0000H.

Figure 12-7. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W

FFF11H (SDR00) FFF10H (SDR00)

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

SDRmn[15:9]							Transfer clock setting by dividing the operating clock (f_{MCK})
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

Cautions 1. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART0 is used.

2. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remarks 1. For the function of the lower 9 bits of the SDRmn register, see **12.2 Configuration of Serial Array Unit**.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.3.6 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 12-8. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00), F010AH, F010BH (SIR01) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn^{Note}	PECTmn	OVCTmn

FECTmn	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PECTmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVCTmn	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVFmn bit of the SSRmn register to 0.

Note The SIR01 register only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00 register) to “0”.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)

2. When the SIRmn register is read, 0000H is always read.

12.3.7 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 12-9. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn[Note]	PEF mn	OVF mn

TSF mn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> ● The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended). ● Communication ends. <Set condition> ● Communication starts.	

BFF mn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
<Clear conditions> ● Transferring transmit data from the SDRmn register to the shift register ends during transmission. ● Reading receive data from the SDRmn register ends during reception. ● The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled). <Set conditions> ● Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode). ● Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). ● A reception error occurs.	

Note The SSR01 register only.

Caution If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVEmn = 1) is detected.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

Figure 12-9. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn^{Note}	PEF mn	OVF mn

FEF mn^{Note}	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear condition> ● 1 is written to the FECTmn bit of the SIRmn register. <Set condition> ● A stop bit is not detected when UART reception ends.	

PEF mn	Parity error detection flag of channel n
0	No error occurs.
1	Parity error occurs (during UART reception).
<Clear condition> ● 1 is written to the PECTmn bit of the SIRmn register. <Set condition> ● The parity of the transmit data and the parity bit do not match when UART reception ends (parity error).	

OVF mn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> ● 1 is written to the OVCTmn bit of the SIRmn register. <Set condition> ● Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). ● Transmit data is not ready for slave transmission or transmission and reception in Simplified SPI (CSI) mode.	

Note The SSR01 register only.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.3.8 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 12-10. Format of Serial Channel Start Register m (SSm)

Address: F0122H, F0123H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS01	SS00

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEmn bit to 1 and enters the communication wait status Note.

Note If set the SSmn = 1 to during a communication operation, will wait status to stop the communication. At this time, holding status value of control register and shift register, SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Cautions 1. Be sure to clear bits 15 to 2 to “0”.

2. For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{MCK} clocks have elapsed.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)

2. When the SSm register is read, 0000H is always read.

12.3.9 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 12-11. Format of Serial Channel Stop Register m (STm)

Address: F0124H, F0125H After reset: 0000H 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	0	0	ST01	ST00

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operation^{Note}.

Note Holding status value of the control register and shift register, the SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution Be sure to clear bits 15 to 2 to "0".

Remarks **1.** m: Unit number (m = 0), n: Channel number (n = 0, 1)

2. When the STm register is read, 0000H is always read.

12.3.10 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 12-12. Format of Serial Channel Enable Status Register m (SEm)

Address: F0120H, F0121H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE01	SE00

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.3.11 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 12-13. Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE 01	SOE 00

SOE mn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution Be sure to clear bits 15 to 2 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.3.12 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 12-14. Format of Serial Output Register m (SOm)

Address: F0128H, F0129H After reset: 0303H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	CKO 01	CKO 00	0	0	0	0	0	0	SO 01	SO 00

CKO mn	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

SO mn	Serial data output of channel n
0	Serial data output value is “0”.
1	Serial data output value is “1”.

Caution Be sure to clear bits 15 to 10 and 7 to 2 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.3.13 Serial output level register m (SOLm)

The SOLm register is a register that is used to set inversion of the data output level of each channel.

This register can be set only in the UART mode. Be sure to set 0 for corresponding bit in the Simplified SPI (CSI) mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting the SOLm register is prohibited when the register is in operation (when SEmn = 1).

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

The lower 8 bits of the SOLm register can be set with an 8-bit memory manipulation instruction with SOLmL.

Reset signal generation clears the SOLm register to 0000H.

Figure 12-15. Format of Serial Output Level Register m (SOLm)

Address: F0134H, F0135H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL 00

SOL mn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

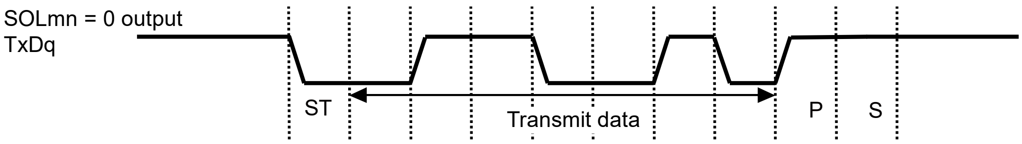
Caution Be sure to clear bits 15 to 1 to "0".

Remark m: Unit number (m = 0), n: Channel number (n = 0)

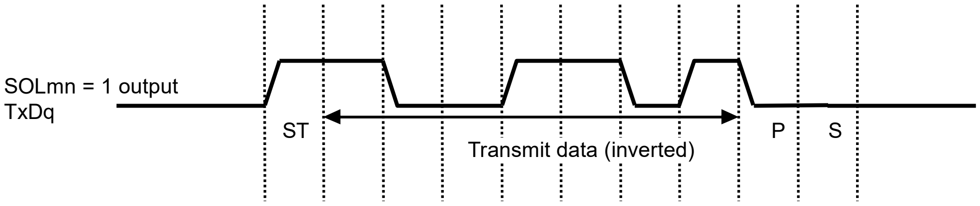
Figure 12-16 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 12-16. Examples of Reverse Transmit Data

(a) Non-reverse Output (SOLmn = 0)



(b) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0), n: Channel number (n = 0)

12.3.14 Serial standby control register m (SSCm)

The SSCm register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI00 or UART0 serial data.

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

The lower 8 bits of the SSCm register can be set with an 8-bit memory manipulation instruction with SSCmL.

Reset signal generation clears the SSCm register to 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.

- **When using CSI00 : Up to 1 Mbps**
- **When using UART0 : 4800 bps only**

Figure 12-17. Format of Serial Standby Control Register m (SSCm)

Address: F0138H 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 transfer end interrupt
0	Enable the generation of error interrupts (INTSRE0/INTSRE2).
1	Disable the generation of error interrupts (INTSRE0/INTSRE2).
● The SSECm bit can be set to 1 or 0 only when both the SWCm and EOCmn bits are set to 1 during UART reception in the SNOOZE mode. In other cases, clear the SSECm bit to 0. ● Setting SSECm, SWCm = 1, 0 is prohibited.	

SWCm	Setting of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.
● When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode). ● The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited. ● Even when using SNOOZE mode, be sure to set the 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.	

Remark m: Unit number (m = 0)

Figure 12-18. 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.

12.3.15 Noise filter enable register 0 (NFEN0)

The NFEN0 register is used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for Simplified SPI(CSI) communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is enabled, CPU/peripheral hardware clock (f_{CLK}) is synchronized with 2-clock match detection. When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock (f_{MCK}).

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 12-19. Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0070H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	0	0	0	0	SNFEN00

SNFEN00	Use of noise filter of RxD0 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN00 bit to 1 to use the RxD0 pin. Clear the SNFEN00 bit to 0 to use the other than RxD0 pin.	

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

12.3.16 Registers controlling port functions of serial input/output pins

Using the serial array unit requires setting of the registers that control the port functions multiplexed on the target channel (port mode register (PMxx), port register (Pxx), port input mode register (PIMxx), port output mode register (POMxx), port mode control register (PMCxx)).

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, **4.3.4 Port input mode register (PIM1)**, **4.3.5 Port output mode register (POM1)**, and **4.3.6 Port mode control registers (PMCxx)**.

Specifically, using a port pin with a multiplexed serial data or serial clock output function (e.g. P12/SO00/TxD0/TOOLTxD/ SEG30) for serial data or serial clock output, requires setting the corresponding bits in the port mode control register (PMCxx) and port mode register (PMxx) to 0, and the corresponding bit in the port register (Pxx) to 1.

When using the port pin in N-ch open-drain output (V_{DD} tolerance/EV_{DD} tolerance) mode, set the corresponding bit in the port output mode register (POMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.5 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Example: When P12/SO00/TxD0/TOOLTxD/SEG30 is to be used for serial data output
Set the PMC12 bit of port mode control register 1 to 0.
Set the PM12 bit of port mode register 1 to 0.
Set the P12 bit of port register 1 to 1.

Specifically, using a port pin with a multiplexed serial data or serial clock input function (e.g. P11/SI00/RxD0/TOOLRxD/ SEG29) for serial data or serial clock input, requires setting the corresponding bit in the port mode register (PMxx) to 1, and the corresponding bit in the port mode control register (PMCxx) to 0. In this case, the corresponding bit in the port register (Pxx) can be set to 0 or 1.

When the TTL input buffer is selected, set the corresponding bit in the port input mode register (PIMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.5 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Example: When P11/SI00/RxD0/TOOLRxD/SEG29 is to be used for serial data input
Set the PMC11 bit of port mode control register 1 to 0.
Set the PM11 bit of port mode register 1 to 1.
Set the P11 bit of port register 1 to 0 or 1.

12.4 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.

12.4.1 Stopping the operation by units

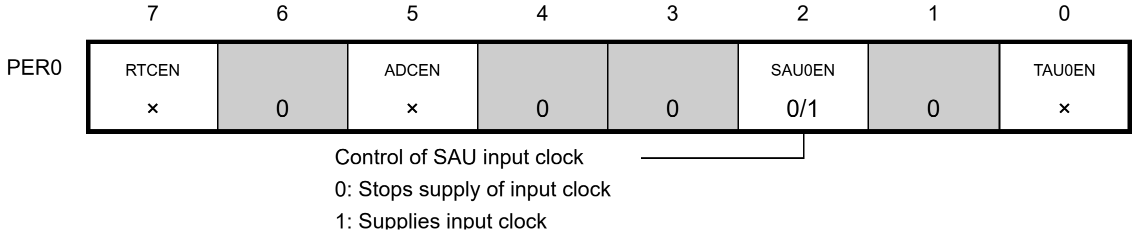
The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

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, set bit 2 (SAU0EN) to 0.

Figure 12-20. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAUm to be stopped to 0.



	7	6	5	4	3	2	1	0
PER0	RTCEN ×	0	ADCEN ×	0	0	SAU0EN 0/1	0	TAU0EN ×

Control of SAU input clock
0: Stops supply of input clock
1: Supplies input clock

Cautions 1. If SAU0EN = 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.

- **Noise filter enable register 0 (NFEN0)**
- **Serial standby control register 0 (SSC0)**
- **Port input mode register 1 (PIM1)**
- **Port output mode register 1 (POM1)**
- **LCD port function registers 0, 3 (PFSEG0, PFSEG3)**
- **Port mode register 1 (PM1)**
- **Port register 1 (P1)**

2. Be sure to clear bits to 6, 4, 3, 1 to “0”.

Remark ▨: Setting disabled (set to the initial value)

×: Bits not used with serial array units (depending on the settings of other peripheral functions)

0/1: Set to 0 or 1 depending on the usage of the user

12.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 12-21. Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm) … This register is a trigger register that is used to enable stopping communication/count by each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	STm1 0/1	STm0 0/1

1: Clears the SEmn bit to 0 and stops the communication operation

* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

(b) Serial Channel Enable Status Register m (SEm) … This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SEm1 0/1	SEm0 0/1

0: Operation stops

* The SEm register is a read-only status register, whose operation is stopped by using the STm register. With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output enable register m (SOEm) … This register is a register that is used to enable or stop output of the serial communication operation of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software.

(d) Serial output register m (SOm) …This register is a buffer register for serial output of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

1: Serial clock output value is “1”

1: Serial data output value is “1”

* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to “1”.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)

2. ▨ : Setting disabled (set to the initial value), 0/1: Set to 0 or 1 depending on the usage of the user

12.5 Operation of Simplified SPI (CSI00, CSI01) 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
  During master communication (CSI00): Max. $f_{MCK}/2$ **Notes 1, 2**
  During master communication (CSI01): Max. $f_{MCK}/4$ **Note 2**
  During slave communication: Max. $f_{MCK}/6$ **Note 2**

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

Notes 1. In master communication (CSI00), maximum transfer rate become $f_{MCK}/2$ when the following conditions.
- $2.7 \text{ V} \leq EV_{DD} = V_{DD} \leq 5.5 \text{ V}$
- $f_{MCK} \leq 12$ MHz

Other cases, maximum transfer rate become $f_{MCK}/4$.

2. Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics (see **CHAPTER 27 ELECTRICAL SPECIFICATIONS**).

The channels supporting Simplified SPI (CSI00, CSI01) are channels 0 and 1.

Channel	Used as Simplified SPI (CSI)	Used as UART
0	CSI00	UART0 (supporting LIN-bus)
1	CSI01	

Simplified SPI (CSI00, CSI01) performs the following seven types of communication operations.

- Master transmission (See **12.5.1**.)
- Master reception (See **12.5.2**.)
- Master transmission/reception (See **12.5.3**.)
- Slave transmission (See **12.5.4**.)
- Slave reception (See **12.5.5**.)
- Slave transmission/reception (See **12.5.6**.)
- SNOOZE mode function (See **12.5.7**.)

12.5.1 Master transmission

Master transmission is that the R7F0C001G/L, R7F0C002G/L outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0	Channel 1
Pins used	SCK00, SO00	SCK01, SO01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	7 or 8 bits	
Transfer rate	Max. $f_{MCK}/2$ [Hz] (CSI00), $f_{MCK}/4$ [Hz] (CSI01) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] Note f_{CLK}: System clock frequency	
Data phase	Selectable by the DAPmn bit of the SCRmn register ● DAPmn = 0: Data output starts from the start of the operation of the serial clock. ● DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register ● CKPmn = 0: Non-reverse (data output at the falling edge and data input at the rising edge of SCK) ● CKPmn = 1: Reverse (data output at the rising edge and data input at the falling edge of SCK)	
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 27 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 12-22. Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00, CSI01) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **12.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Transmit data (Transmit data setting)							

SIOp

(d) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	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	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

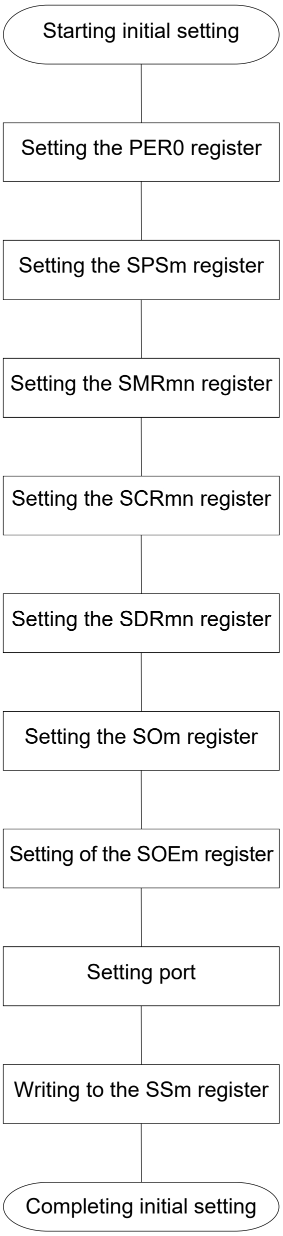
2. ▨: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-23. Initial Setting Procedure for Master Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

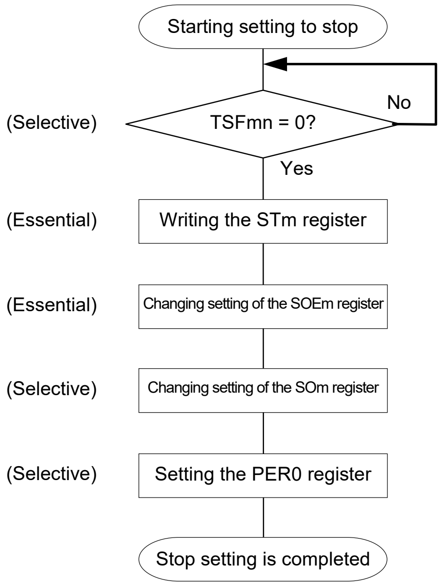
Set the SOEmn bit to 1 and enable data output of the target channel.

Setting a port register and a port mode register (Enable data output and clock output of the target channel by)

Set the SSmn bit of the target channel to 1 (SEmn bit = 1 : to enable operation).

Setting of SAU is completed.
Write transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Figure 12-24. Procedure for Stopping Master Transmission



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write 1 to the STmn bit of the target channel.
(SEmn = 0 : to operation stop status)

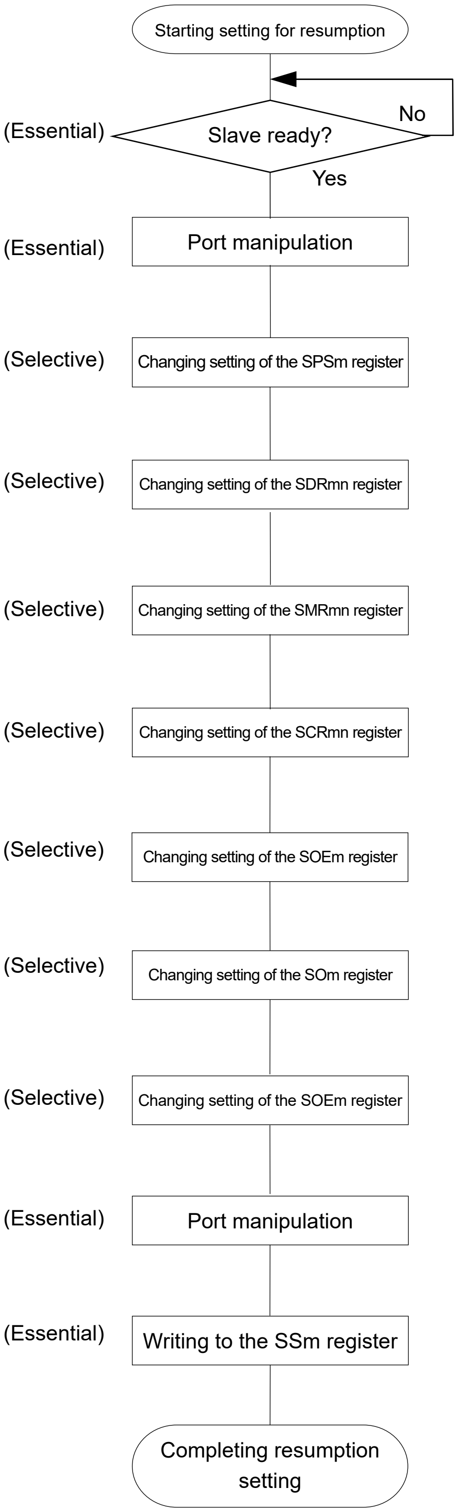
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Figure 12-25. Procedure for Resuming Master Transmission



Wait until stop the communication target (slave) or communication operation completed

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

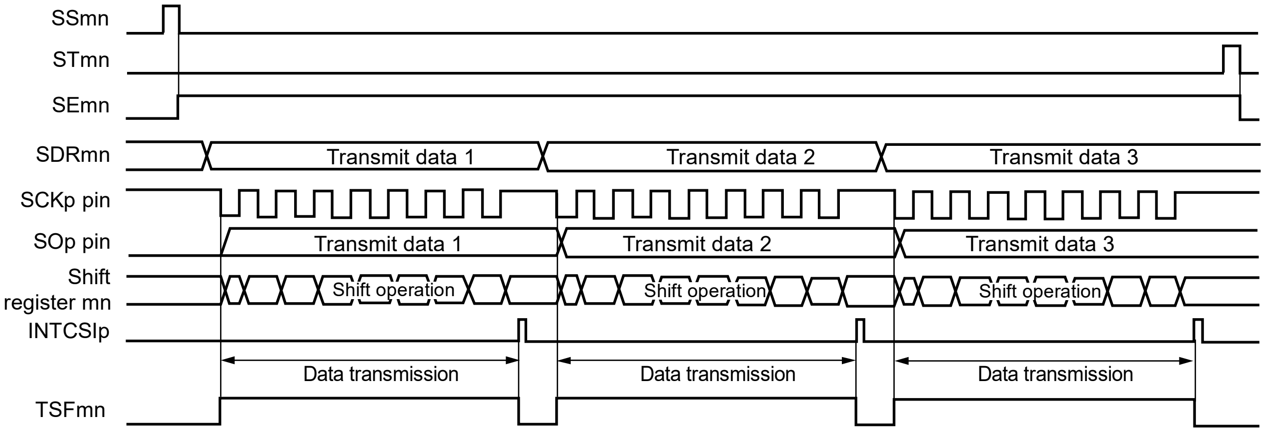
Set the SSmn bit of the target channel to 1 (SEmn = 1 : to enable operation).

Setting is completed
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

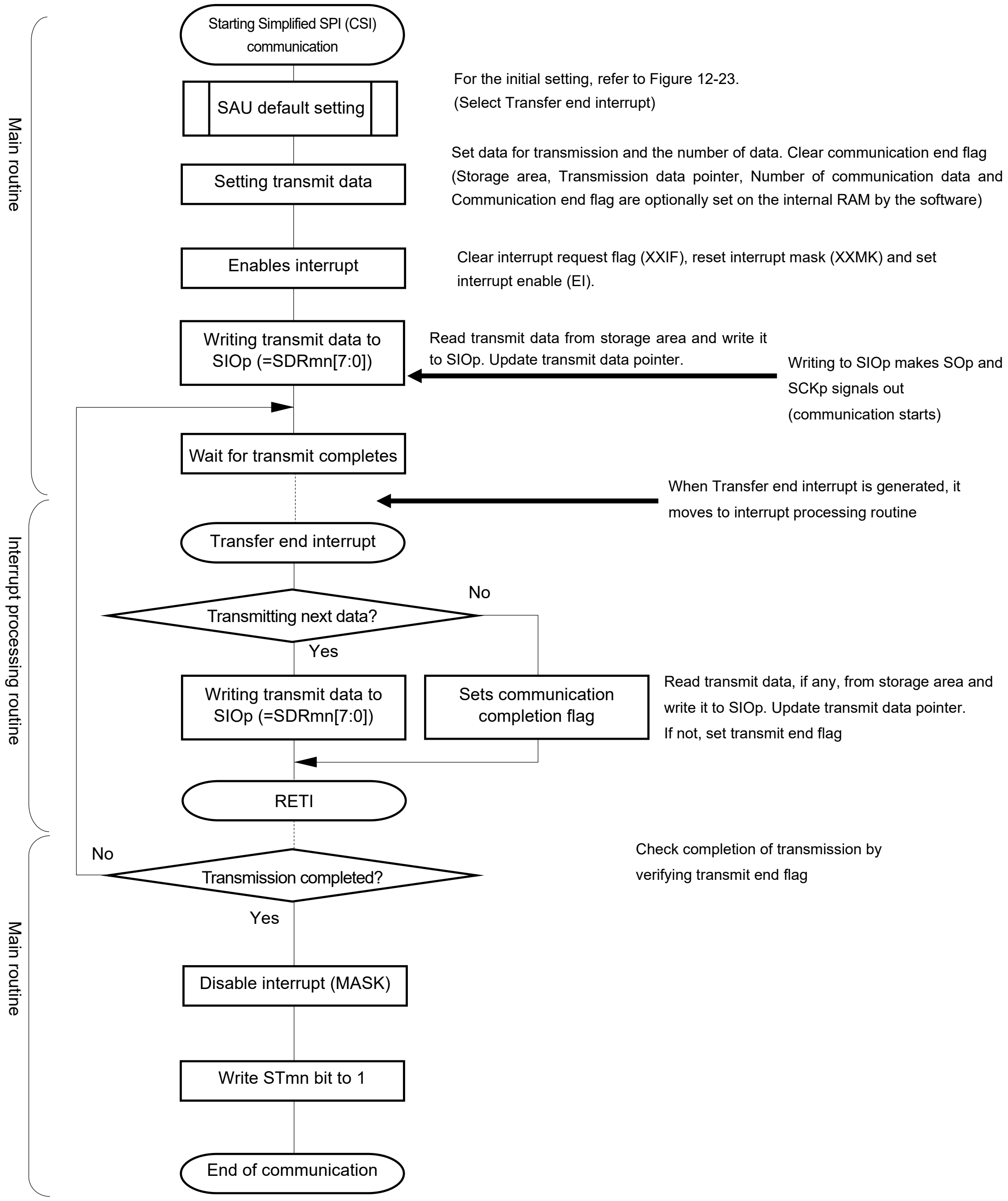
(3) Processing flow (in single-transmission mode)

Figure 12-26. Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 12-27. Flowchart of Master Transmission (in Single-Transmission Mode)



For the initial setting, refer to Figure 12-23.
(Select Transfer end interrupt)

Set data for transmission and the number of data. Clear communication end flag (Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).

Read transmit data from storage area and write it to SIOp. Update transmit data pointer.

Writing to SIOp makes SOp and SCKp signals out (communication starts)

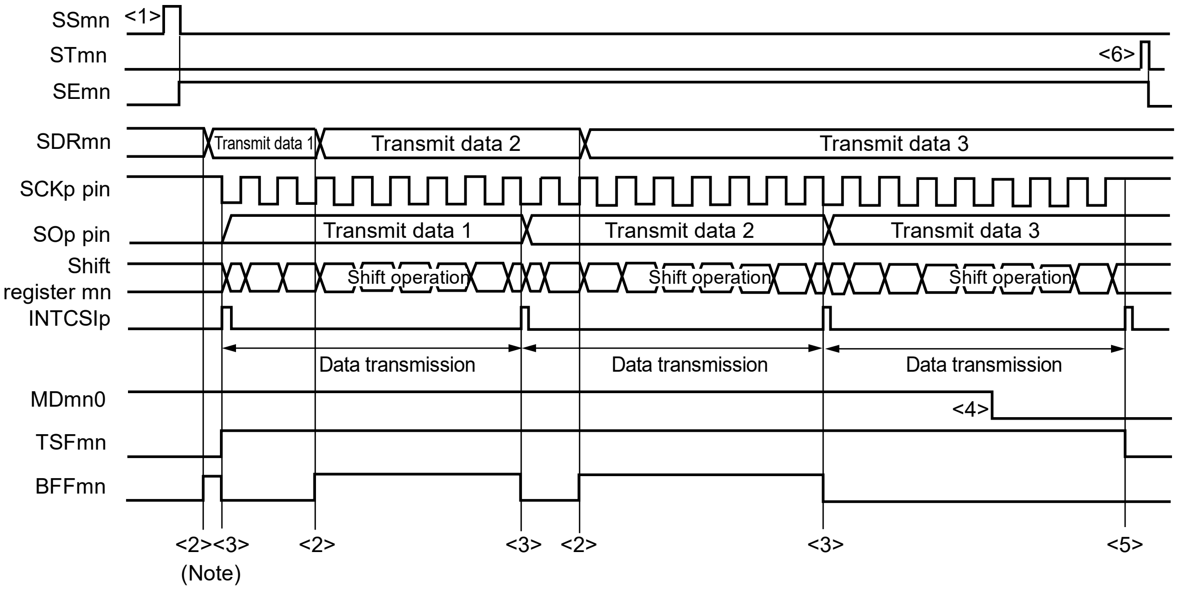
When Transfer end interrupt is generated, it moves to interrupt processing routine

Read transmit data, if any, from storage area and write it to SIOp. Update transmit data pointer.
If not, set transmit end flag

Check completion of transmission by verifying transmit end flag

(4) Processing flow (in continuous transmission mode)

Figure 12-28. Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

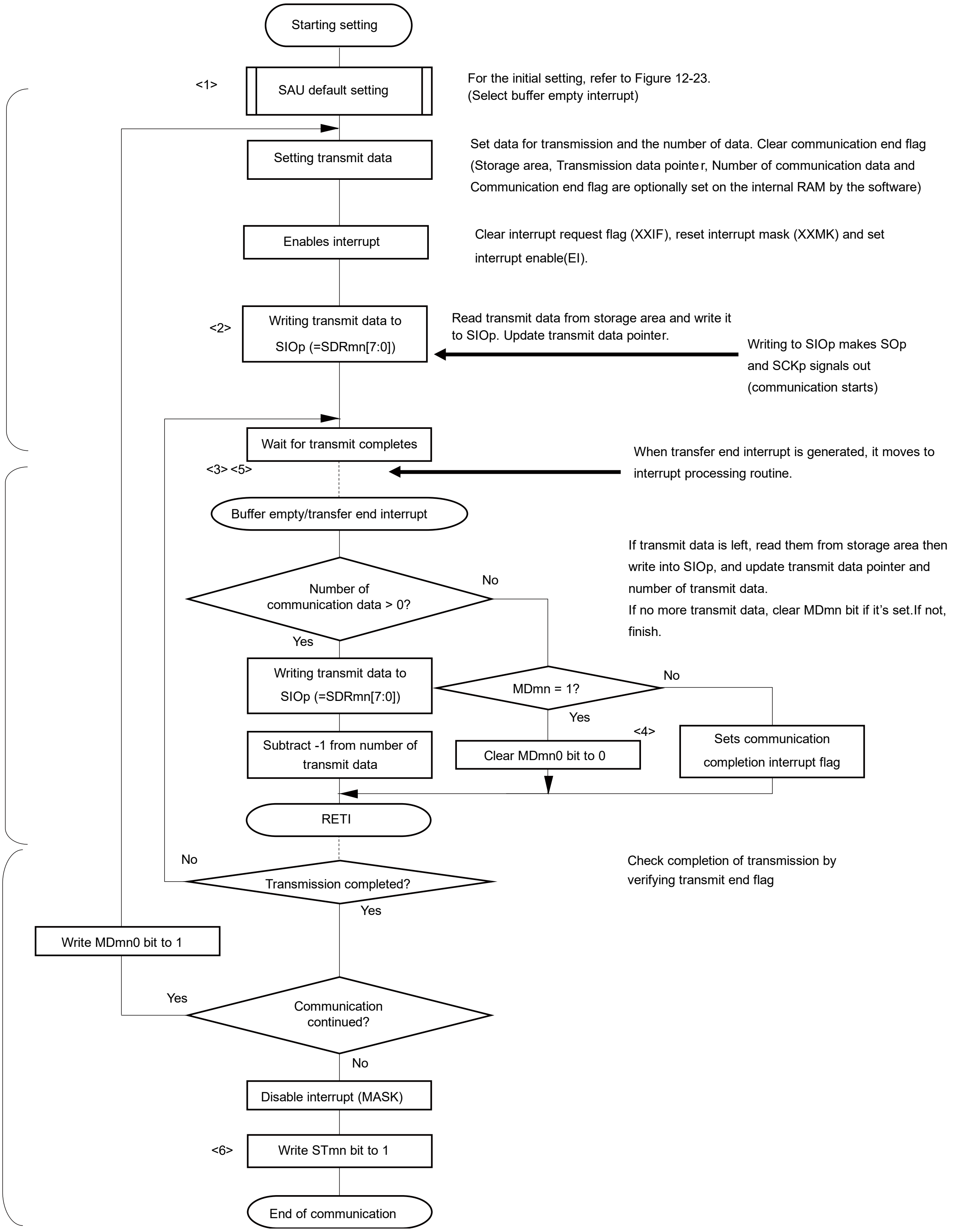


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 12-29. Flowchart of Master Transmission (in Continuous Transmission Mode)



For the initial setting, refer to Figure 12-23.
(Select buffer empty interrupt)

Set data for transmission and the number of data. Clear communication end flag (Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable(EI).

Read transmit data from storage area and write it to SIOp. Update transmit data pointer.

Writing to SIOp makes SOp and SCKp signals out (communication starts)

When transfer end interrupt is generated, it moves to interrupt processing routine.

If transmit data is left, read them from storage area then write into SIOp, and update transmit data pointer and number of transmit data.
If no more transmit data, clear MDmn bit if it's set.If not, finish.

Check completion of transmission by verifying transmit end flag

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-28 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

12.5.2 Master reception

Master reception is that the R7F0C001G/L, R7F0C002G/L a transfer clock and receives data from other device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0	Channel 1
Pins used	SCK00, SI00	SCK01, SI01
Interrupt	INTCSI00	INTCSI01
	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. $f_{MCK}/2$ [Hz] (CSI00), $f_{MCK}/4$ [Hz] (CSI01) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]^{Note} f_{CLK}: System clock frequency	
Data phase	Selectable by the DAPmn bit of the SCRmn register ● DAPmn = 0: Data 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	

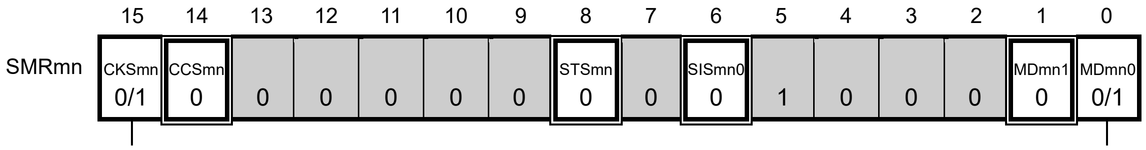
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 27 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 12-30. Example of Contents of Registers for Master Reception of Simplified SPI (CSI00, CSI01) (1/2)

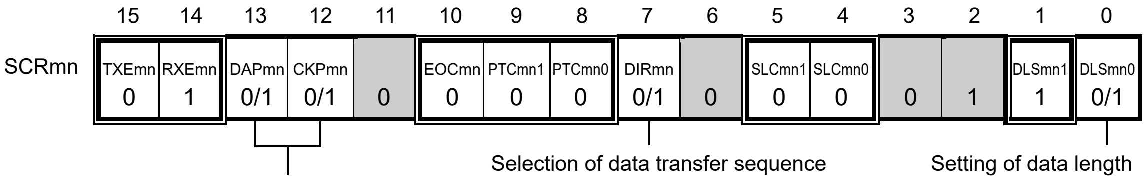
(a) Serial mode register mn (SMRmn)



Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

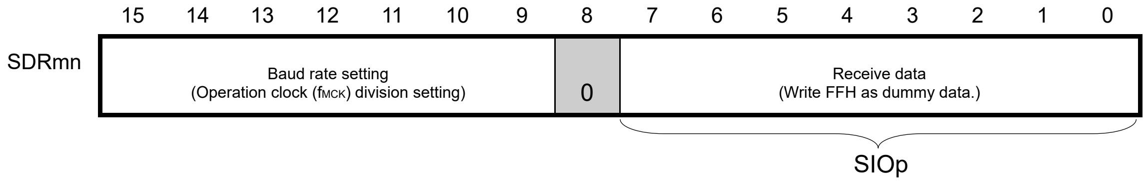


Selection of the data and clock phase (For details about the setting, see **12.3 Registers Controlling Serial Array Unit**.)

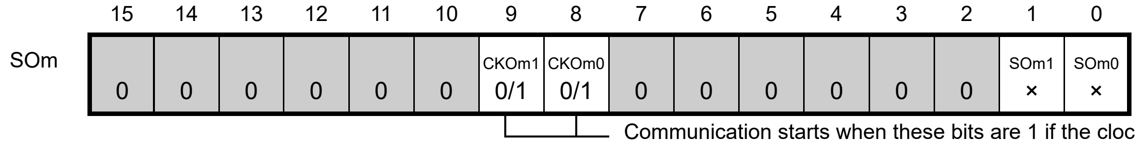
Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) … Sets only the bits of the target channel.



Communication starts when these bits are 1 if the clock phase is non-reverse (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

Figure 12-30. Example of Contents of Registers for Master Reception of Simplified SPI (CSI00, CSI01) (2/2)

(e) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	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	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

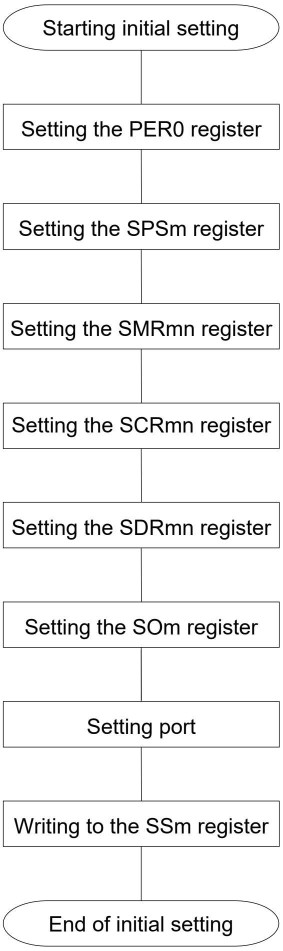
2. ▨ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-31. Initial Setting Procedure for Master Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

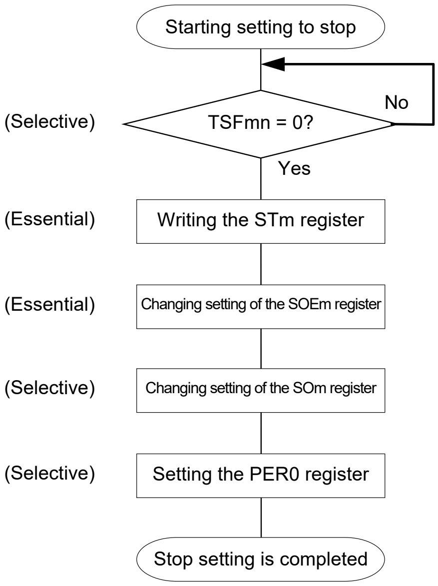
Set the initial output level of the serial clock (CKOmn).

Enable clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Set dummy data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Figure 12-32. Procedure for Stopping Master Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write 1 to the STmn bit of the target channel.
(SEmn = 0 : to operation stop status)

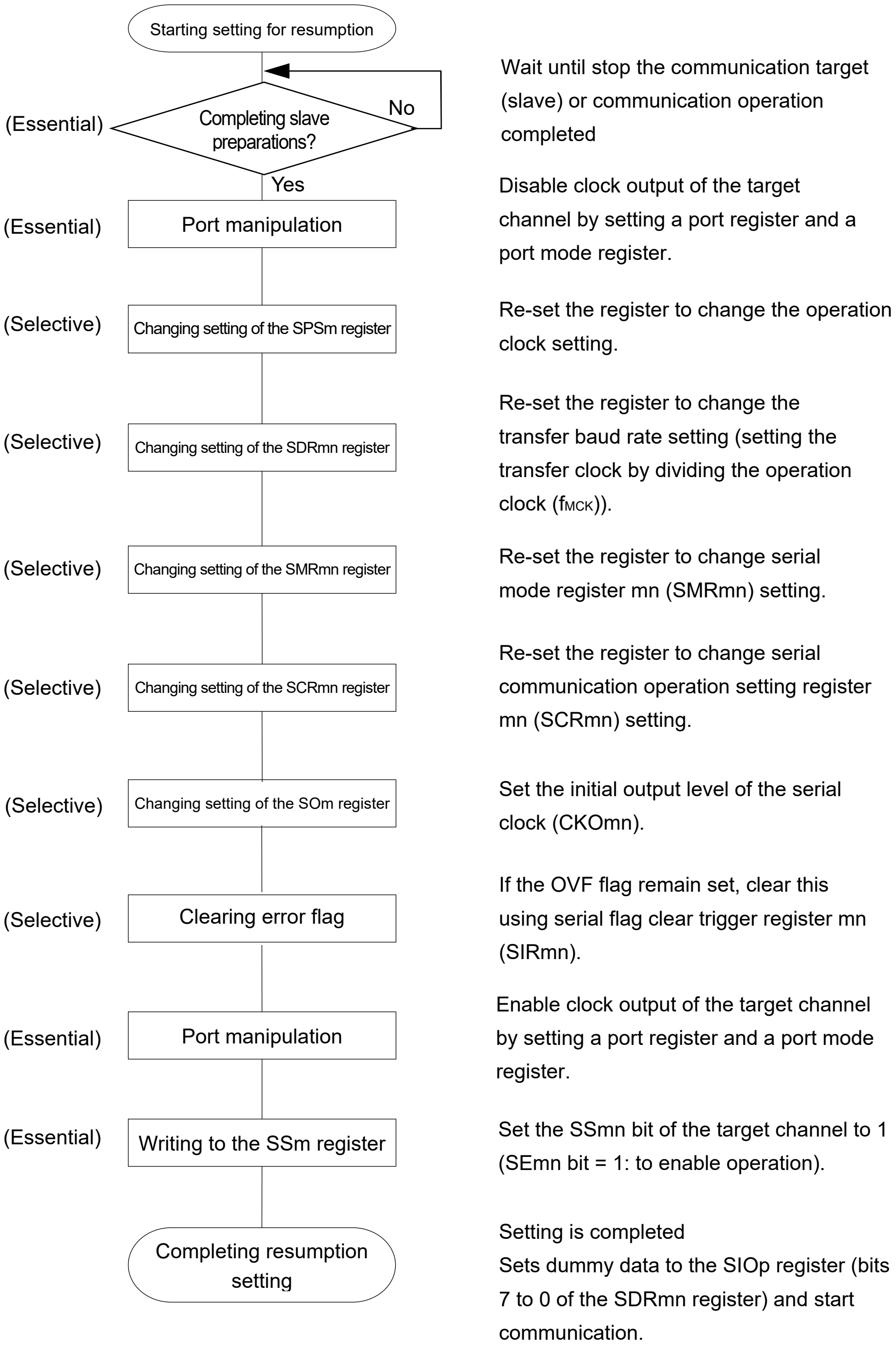
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

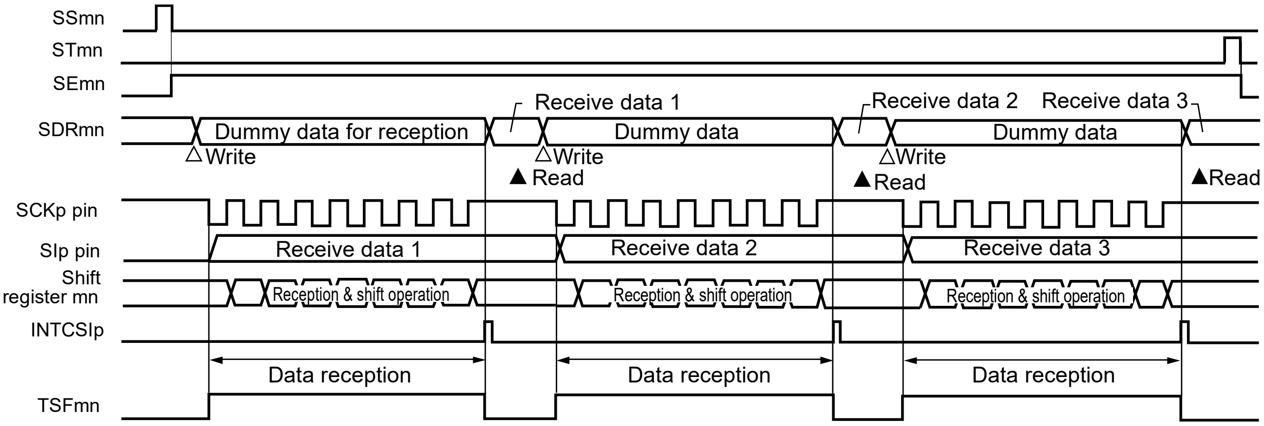
Figure 12-33. Procedure for Resuming Master Reception



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.

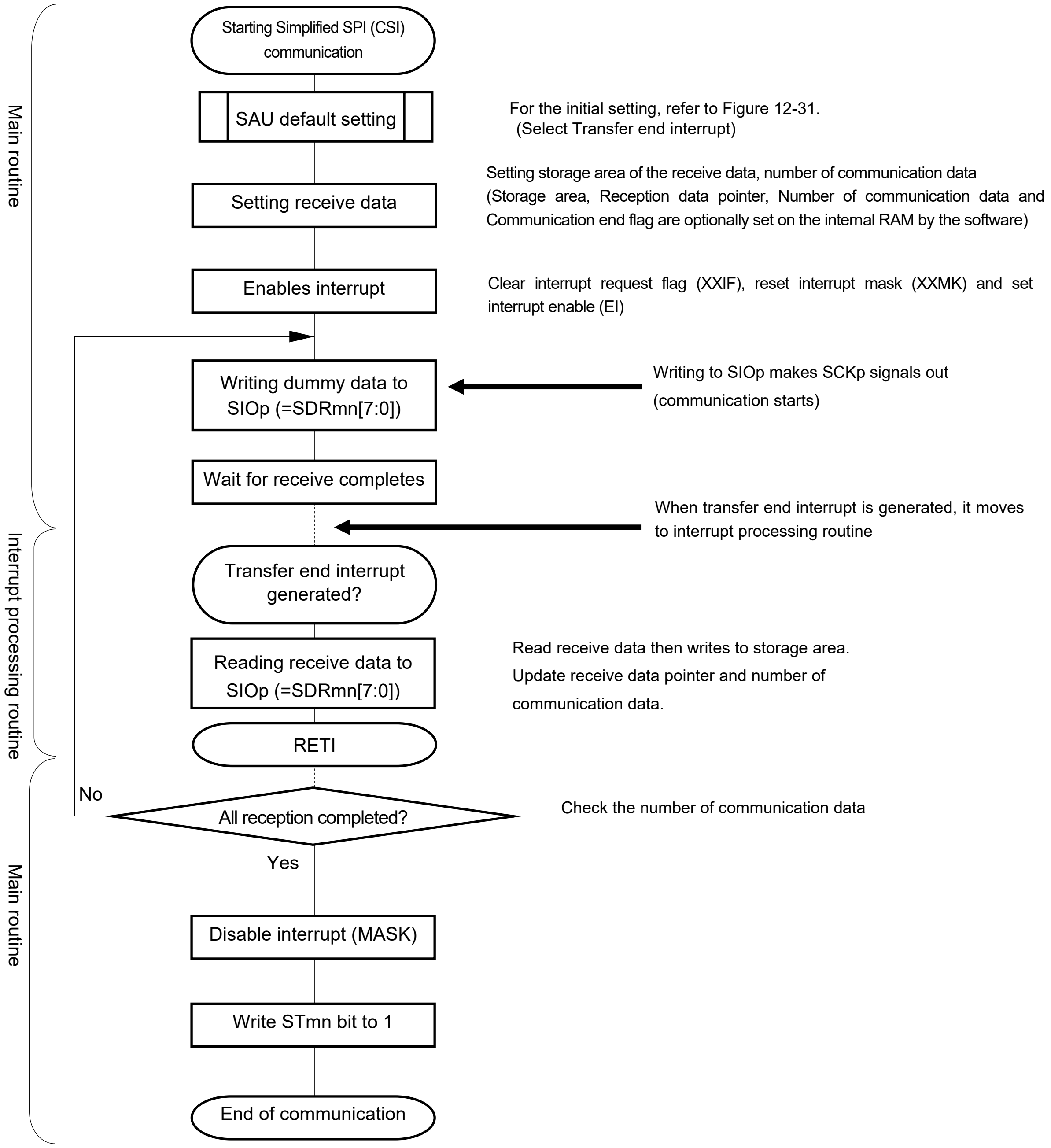
(3) Processing flow (in single-reception mode)

Figure 12-34. Timing Chart of Master Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 12-35. Flowchart of Master Reception (in Single-Reception Mode)



For the initial setting, refer to Figure 12-31.
(Select Transfer end interrupt)

Setting storage area of the receive data, number of communication data (Storage area, Reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)

Writing to SIOp makes SCKp signals out (communication starts)

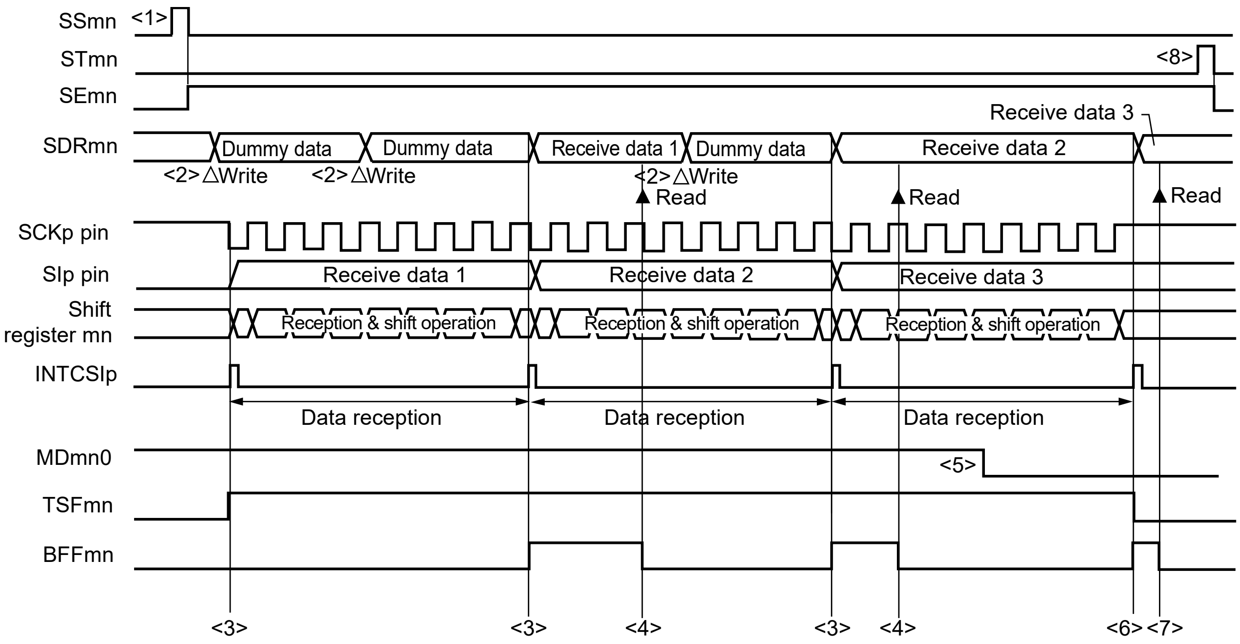
When transfer end interrupt is generated, it moves to interrupt processing routine

Read receive data then writes to storage area. Update receive data pointer and number of communication data.

Check the number of communication data

(4) Processing flow (in continuous reception mode)

Figure 12-36. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

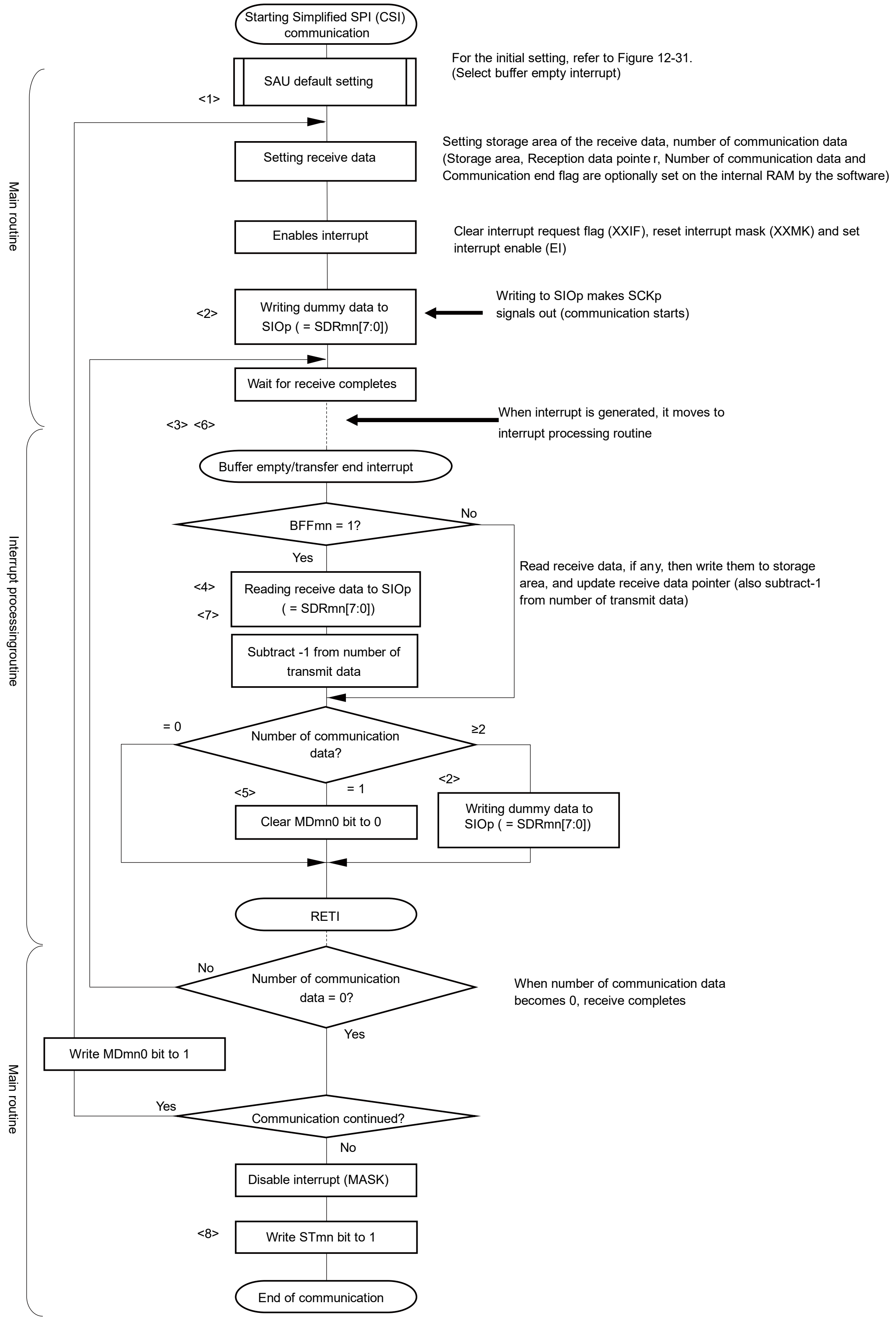


Caution **The MDmn0 bit can be rewritten even during operation.**
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-37 Flowchart of Master Reception (in Continuous Reception Mode)**.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 12-37. Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-36 Timing Chart of Master Reception (in Continuous Reception Mode)**.

12.5.3 Master transmission/reception

Master transmission/reception is that the R7F0C001G/L, R7F0C002G/L outputs a transfer clock and transmits/receives data to/from other device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0	Channel 1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01
Interrupt	INTCSI00	INTCSI01
	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. $f_{MCK}/2$ [Hz] (CSI00), $f_{MCK}/4$ [Hz] (CSI01) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]**Note** f_{CLK}: System clock frequency	
Data phase	Selectable by the DAPmn bit of the SCRmn register ● DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. ● DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register ● CKPmn = 0: Non-reverse ● CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

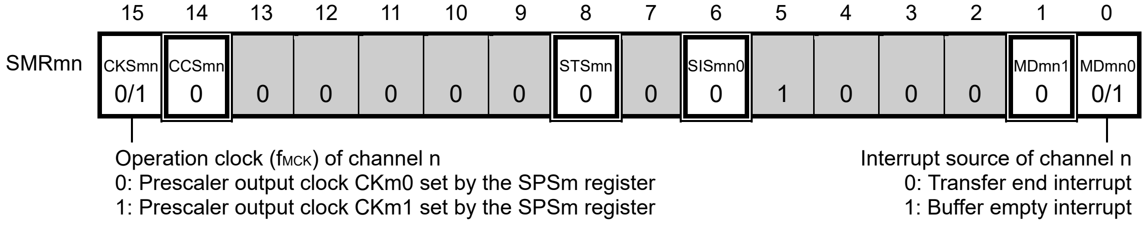
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 27 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 12-38. Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI01) (1/2)

(a) Serial mode register mn (SMRmn)

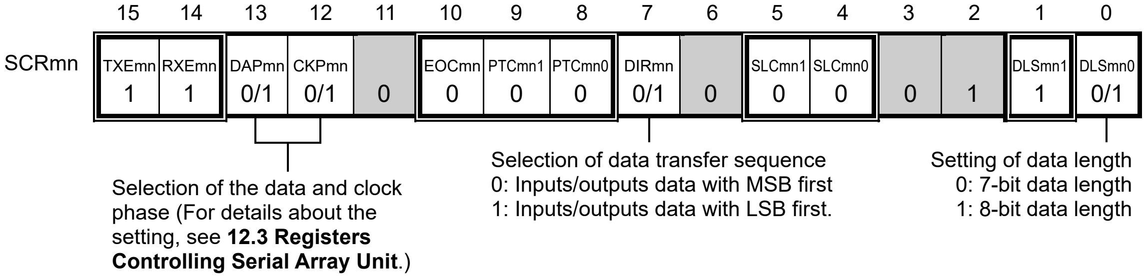


	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	0	MDmn1 0	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)



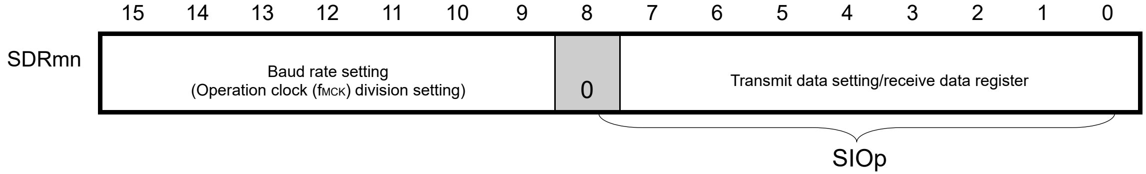
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **12.3 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

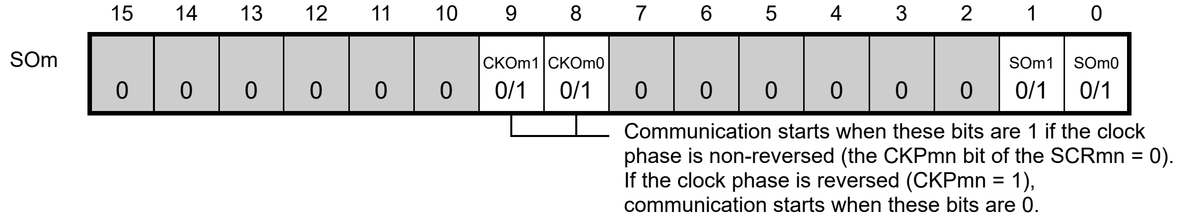
(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



	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	Transmit data setting/receive data register							

SIOp

(d) Serial output register m (SOm) … Sets only the bits of the target channel.



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

2. ☐: Setting is fixed in the Simplified SPI (CSI) master transmission/reception mode

▨ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-38. Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI01) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	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	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

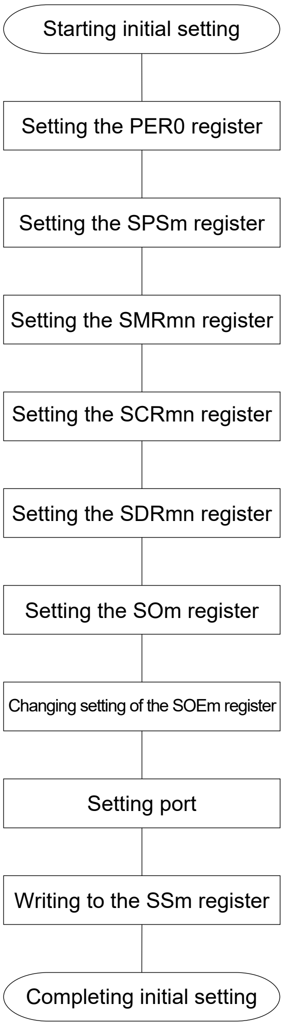
2. ■ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-39. Initial Setting Procedure for Master Transmission/Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

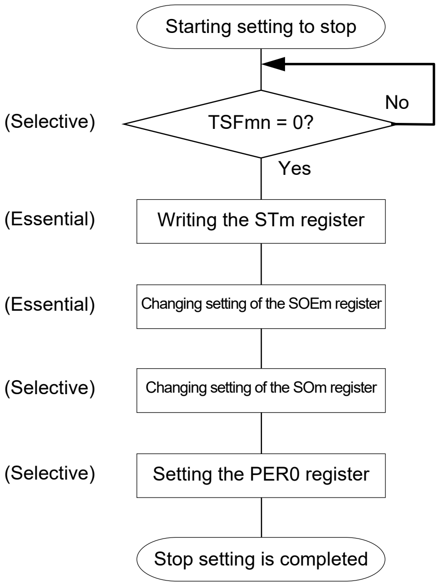
Set the SOEmn bit to 1 and enable data output of the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Figure 12-40. Procedure for Stopping Master Transmission/Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write 1 to the STmn bit of the target channel.
(SEmn = 0 : to operation stop status)

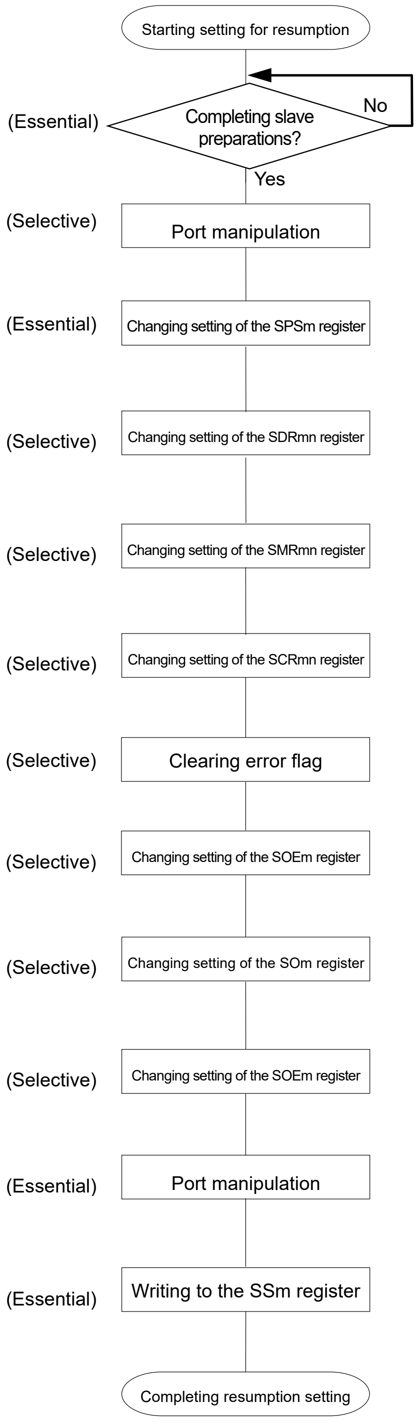
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Figure 12-41. Procedure for Resuming Master Transmission/Reception



Wait until stop the communication target (slave) or communication operation completed

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remain set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

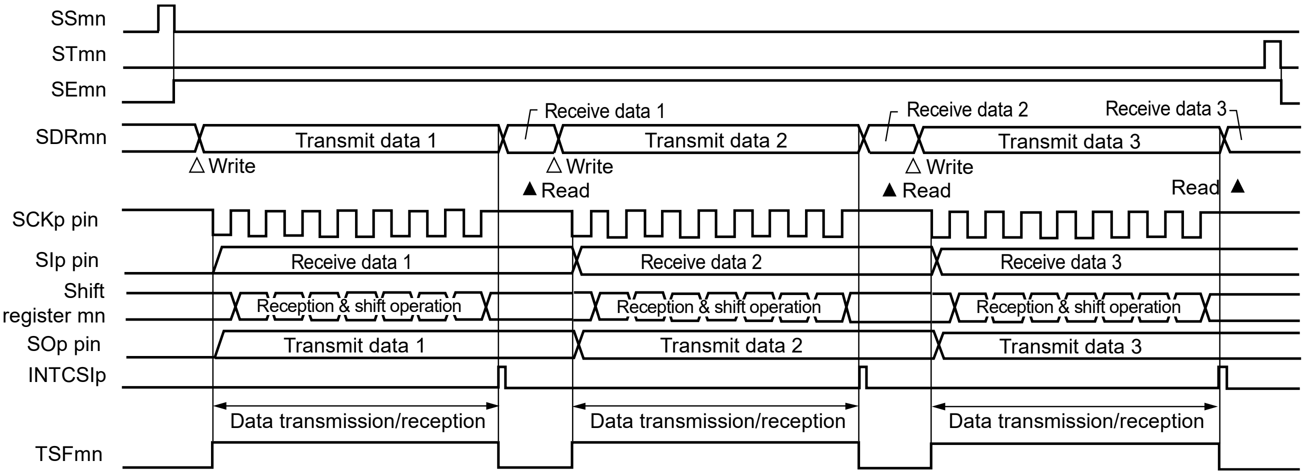
Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn = 1 : to enable operation).

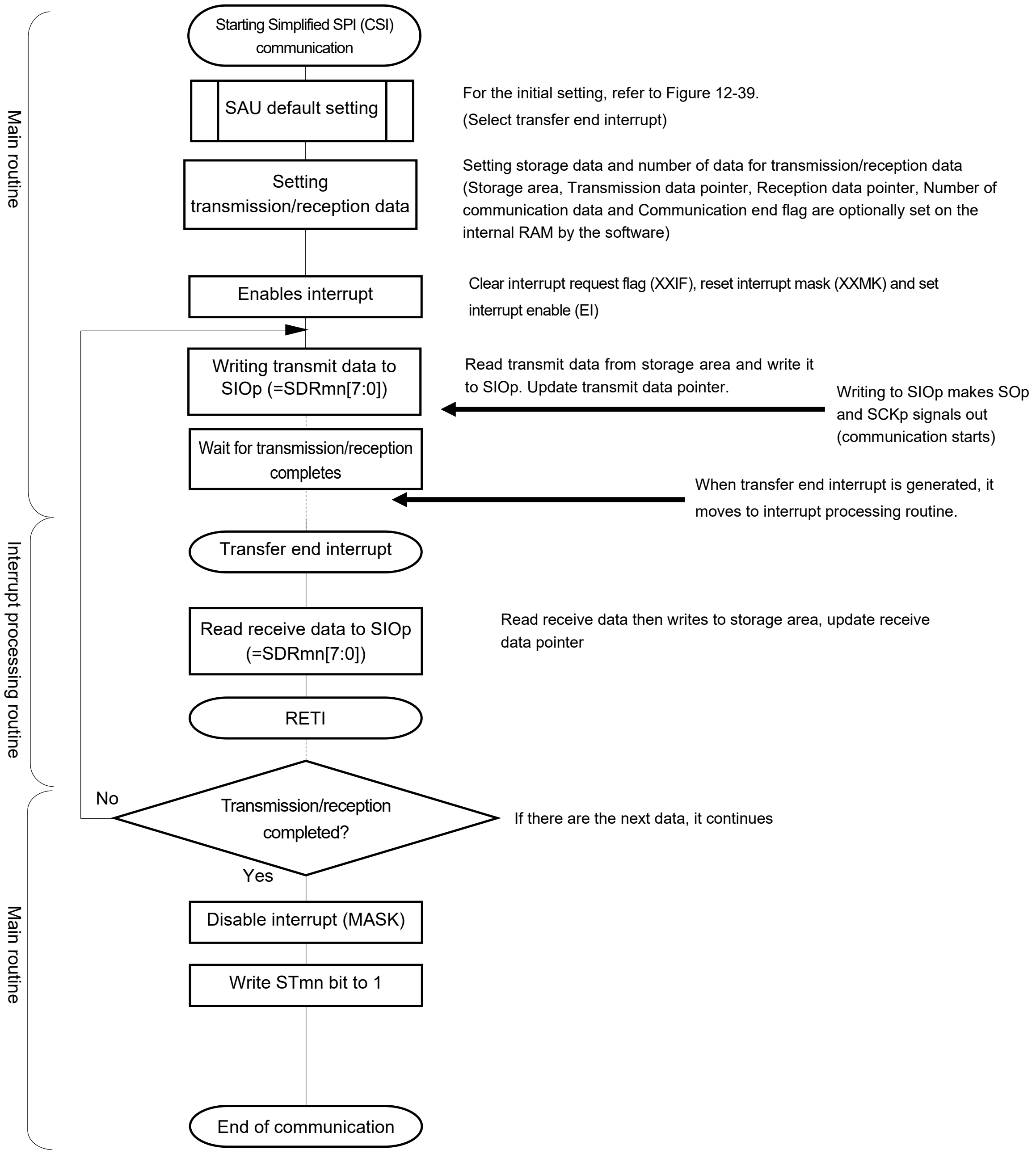
(3) Processing flow (in single-transmission/reception mode)

Figure 12-42. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



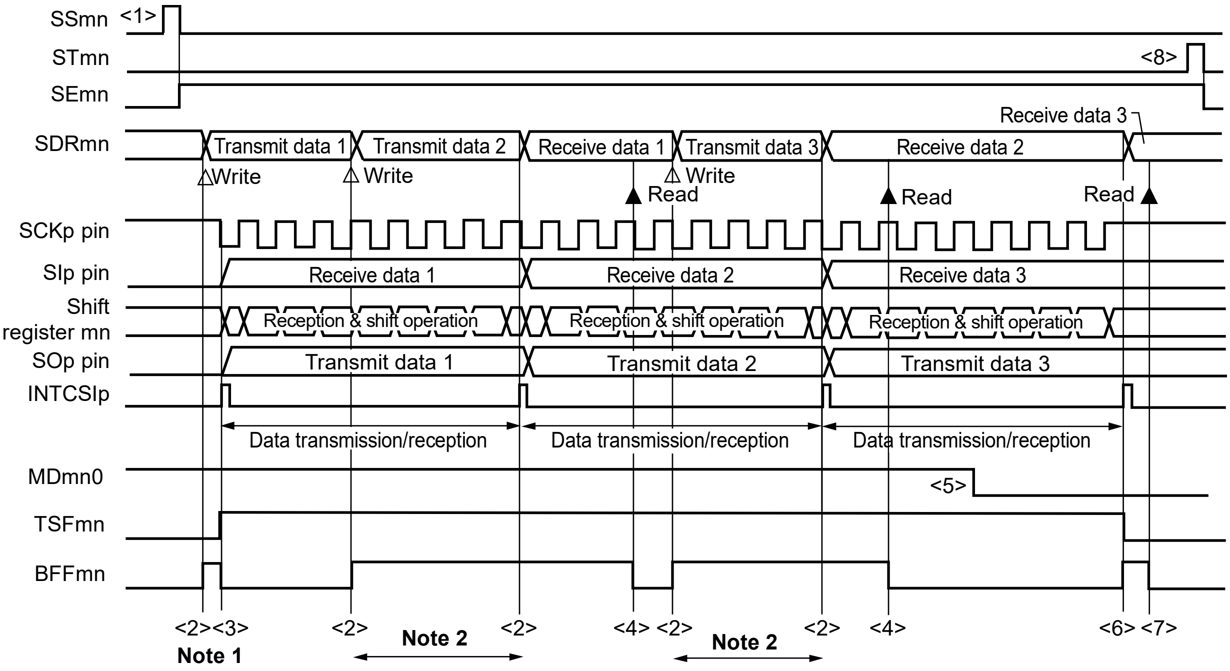
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 12-43. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)



(4) Processing flow (in continuous transmission/reception mode)

Figure 12-44. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

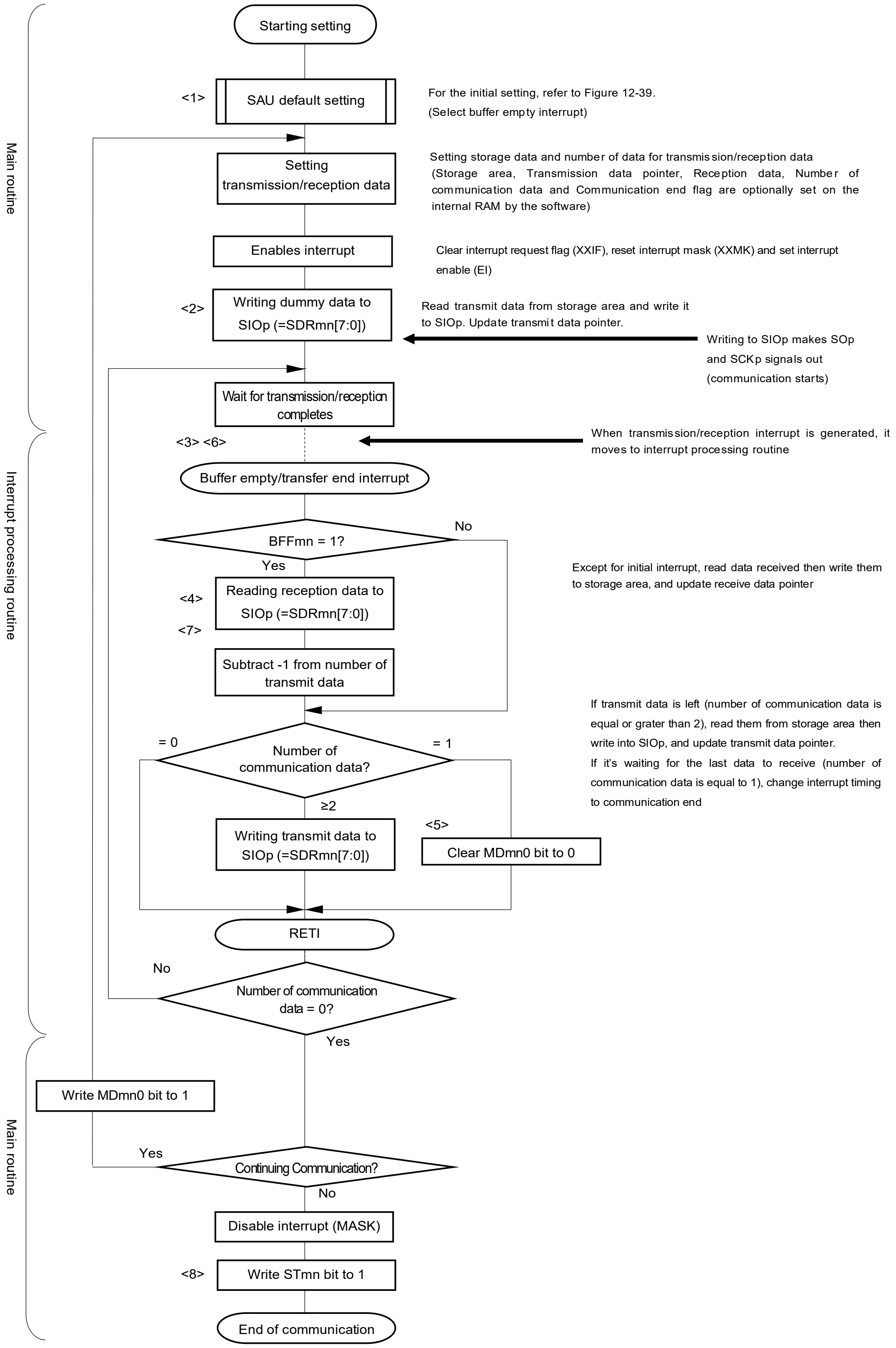
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-45 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 12-45. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-44 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

12.5.4 Slave transmission

Slave transmission is that the R7F0C001G/L, R7F0C002G/L transmits data to another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0	Channel 1
Pins used	SCK00, SO00	SCK01, SO01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. f_{MCK}/6 [Hz][Notes 1, 2].	
Data phase	Selectable by the DAPmn bit of the SCRmn register ● DAPmn = 0: Data output starts from the start of the operation of the serial clock. ● DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register ● CKPmn = 0: Non-reverse ● CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Notes 1. Because the external serial clock input to the SCK00, and SCK01 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 27 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

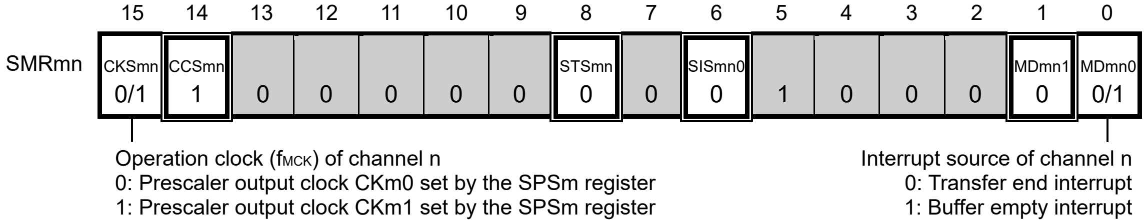
f_{SCK}: Serial clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 12-46. Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI01) (1/2)

(a) Serial mode register mn (SMRmn)

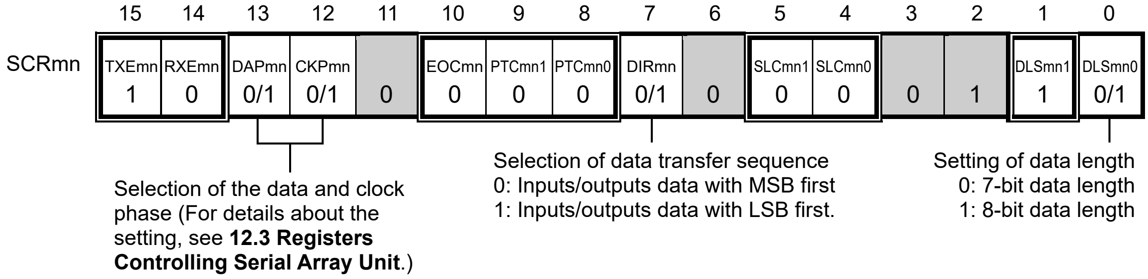


	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0					MDmn1	MDmn0
	0/1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

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)



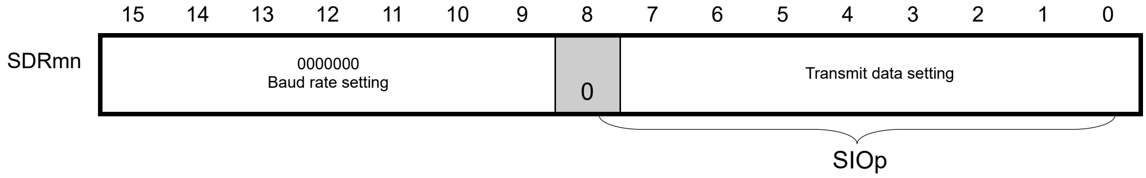
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **12.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

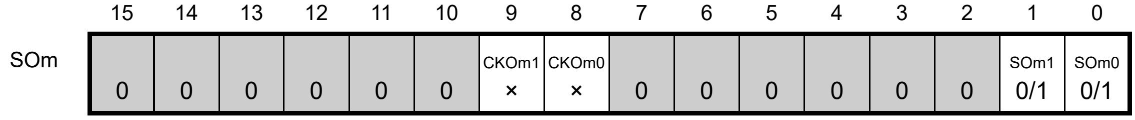
Setting of data length
0: 7-bit data length
1: 8-bit data length

(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							

(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							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	×	×	0	0	0	0	0	0	0/1	0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

2. ☐: Setting is fixed in the Simplified SPI(CSI) slave transmission mode, ▨ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-46. Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI01) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	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	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

2. ▨: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-47. Initial Setting Procedure for Slave Transmission

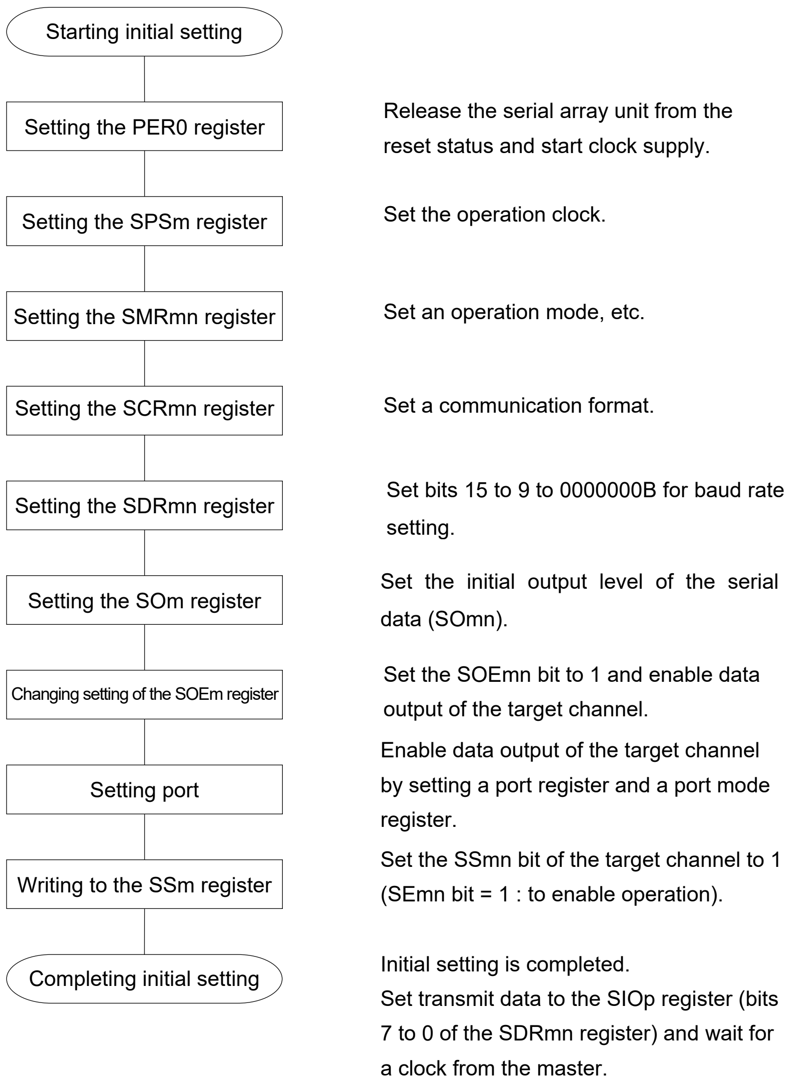
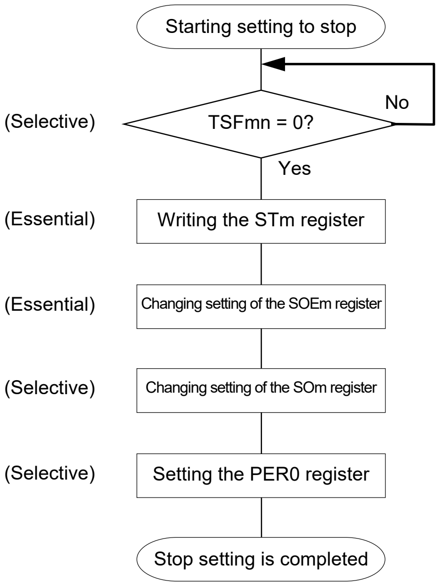


Figure 12-48. Procedure for Stopping Slave Transmission



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write 1 to the STmn bit of the target channel.
(SEmn = 0 : to operation stop status)

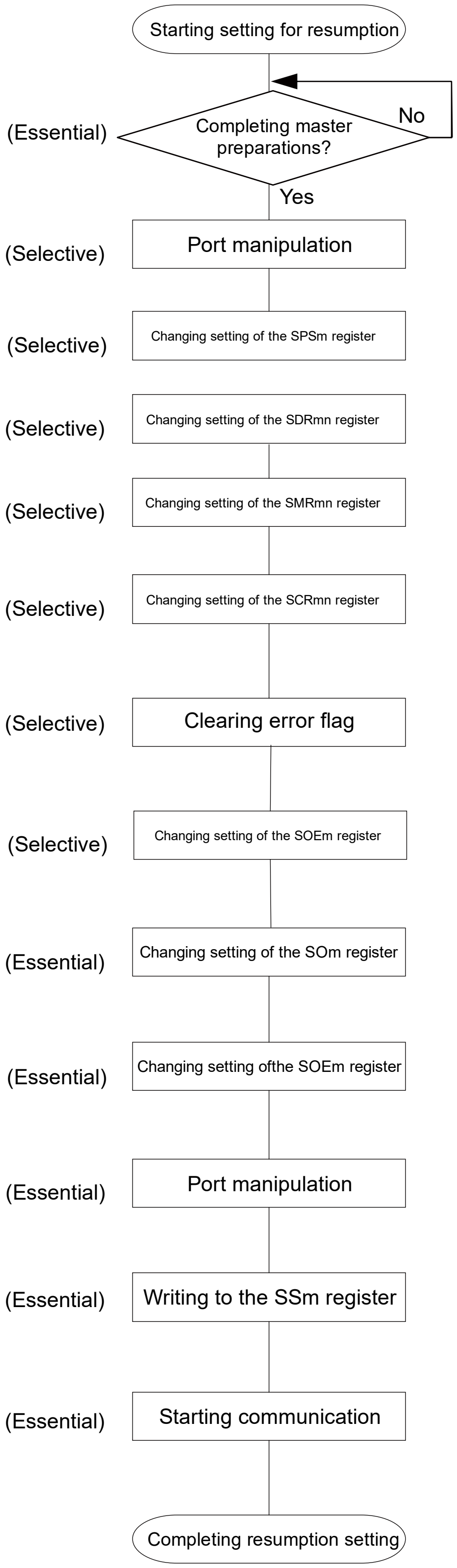
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

To use the STOP mode, reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Figure 12-49. Procedure for Resuming Slave Transmission



Wait until stop the communication target (master)

Disabledata output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remainset, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

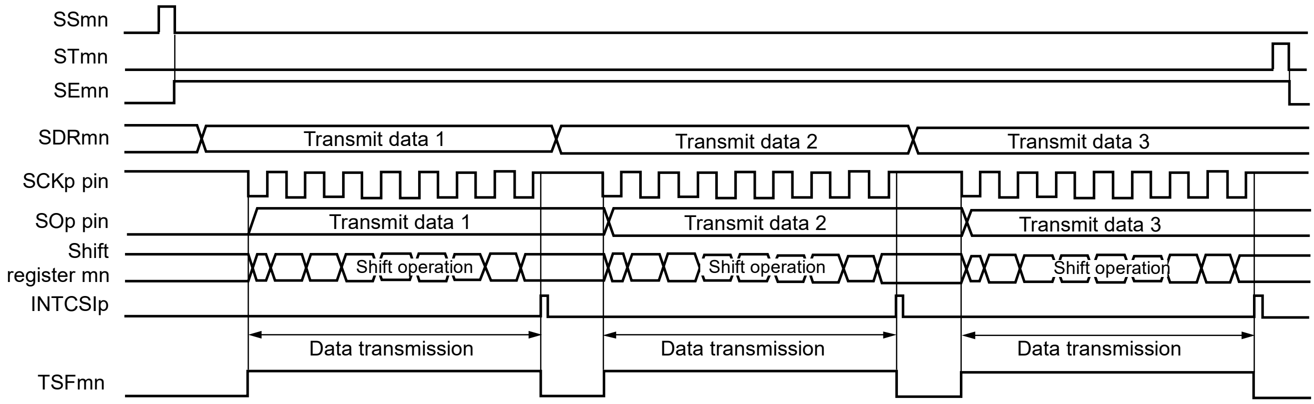
Set the SSmn bit of the target channel to 1 (SEmn = 1 : to enable operation).

Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

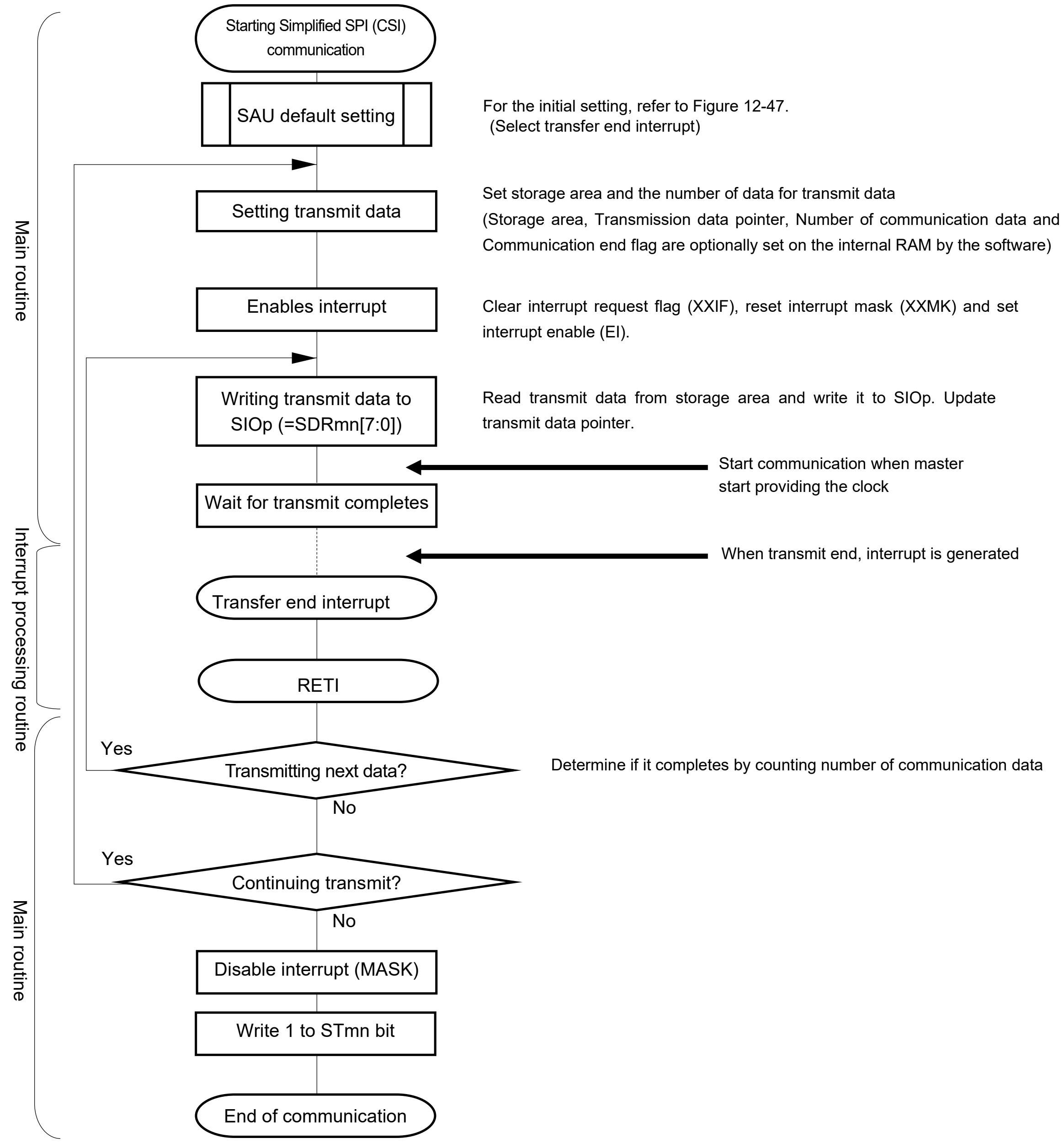
(3) Processing flow (in single-transmission mode)

Figure 12-50. Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



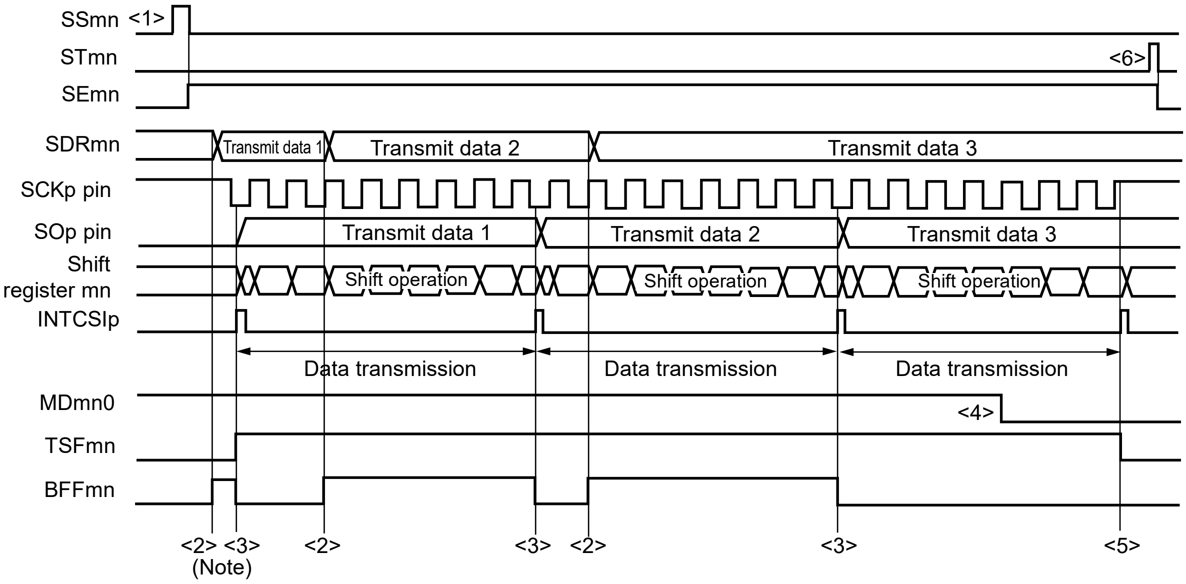
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 12-51. Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 12-52. Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

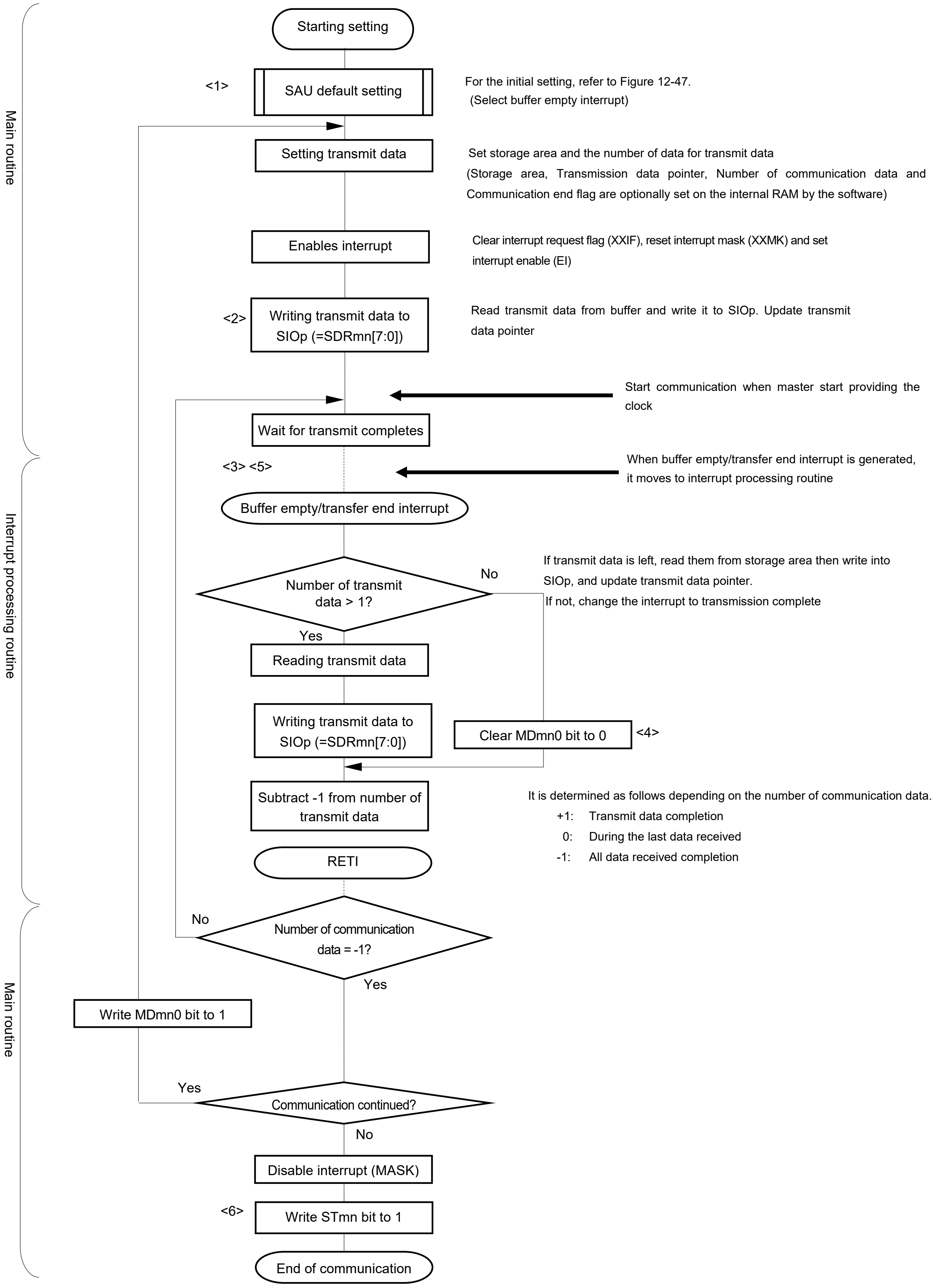


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 12-53. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-52 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

12.5.5 Slave reception

Slave reception is that the R7F0C001G/L, R7F0C002G/L receives data from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0	Channel 1
Pins used	SCK00, SI00	SCK01, SI01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. $f_{MCK}/6$ [Hz] **Notes 1, 2**	
Data phase	Selectable by the DAPmn bit of the SCRmn register ● DAPmn = 0: Data input starts from the start of the operation of the serial clock. ● DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register ● CKPmn = 0: Non-reverse ● CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Notes 1. Because the external serial clock input to the SCK00, and SCK01 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 27 ELECTRICAL SPECIFICATIONS**).

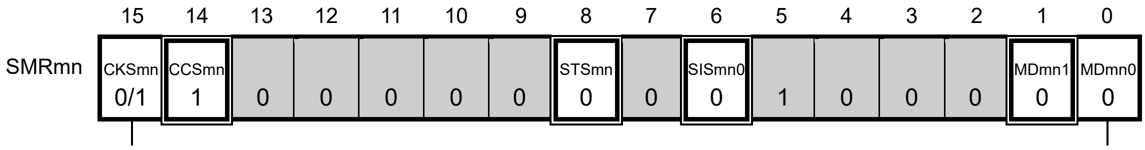
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 12-54. Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01) (1/2)

(a) Serial mode register mn (SMRmn)

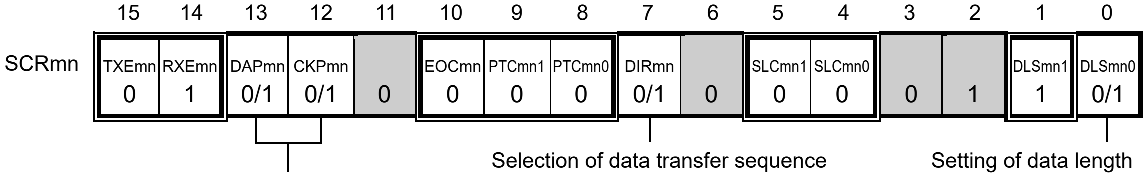


	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	0	MDmn1 0	MDmn0 0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)



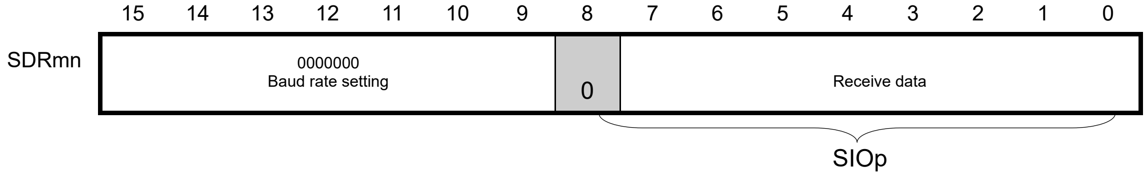
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **12.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

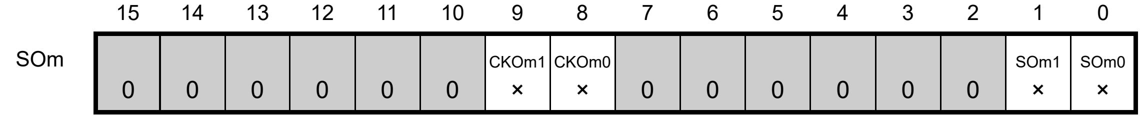
Setting of data length
0: 7-bit data length
1: 8-bit data length

(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							

(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	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 ×

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

2. □: Setting is fixed in the Simplified SPI(CSI) slave transmission mode, ▨: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-54. Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01) (2/2)

(e) Serial output enable register m (SOEm) …The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	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	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

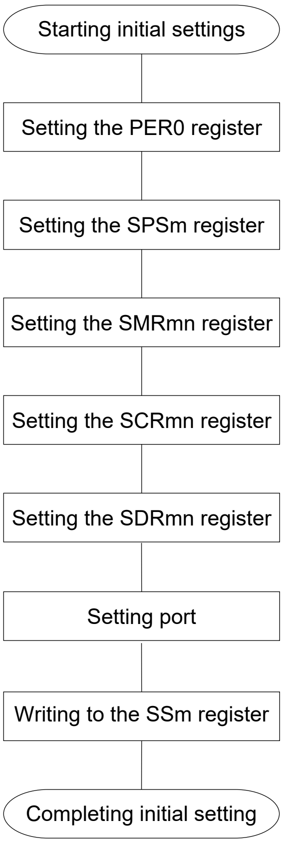
2. ▨ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-55. Initial Setting Procedure for Slave Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

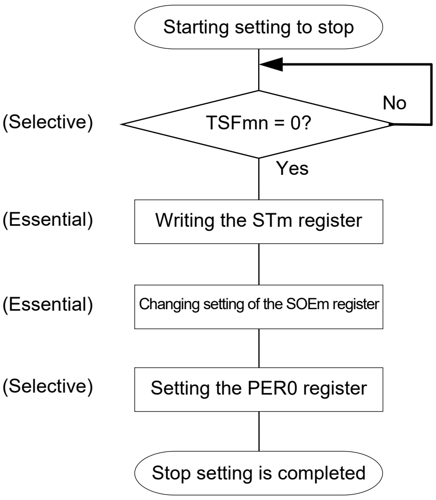
Set a communication format.

Set baud rate setting (bits 15 to 9) to 0000000B.

Enable data input and clock input of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation). Wait for a clock from the master.

Figure 12-56. Procedure for Stopping Slave Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

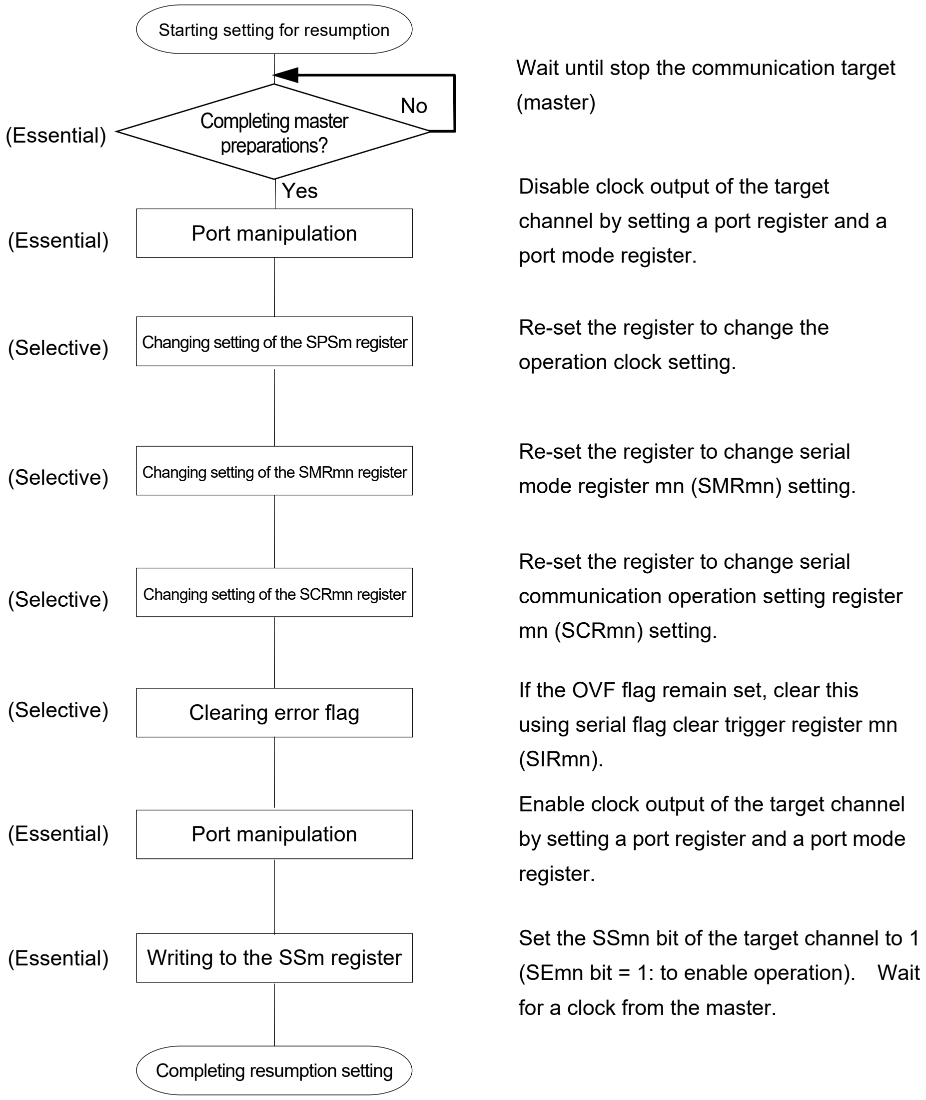
Write 1 to the STmn bit of the target channel.
(SEmn = 0 : to operation stop status)

Set the SOEmn bit to 0 and stop the output of the target channel.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

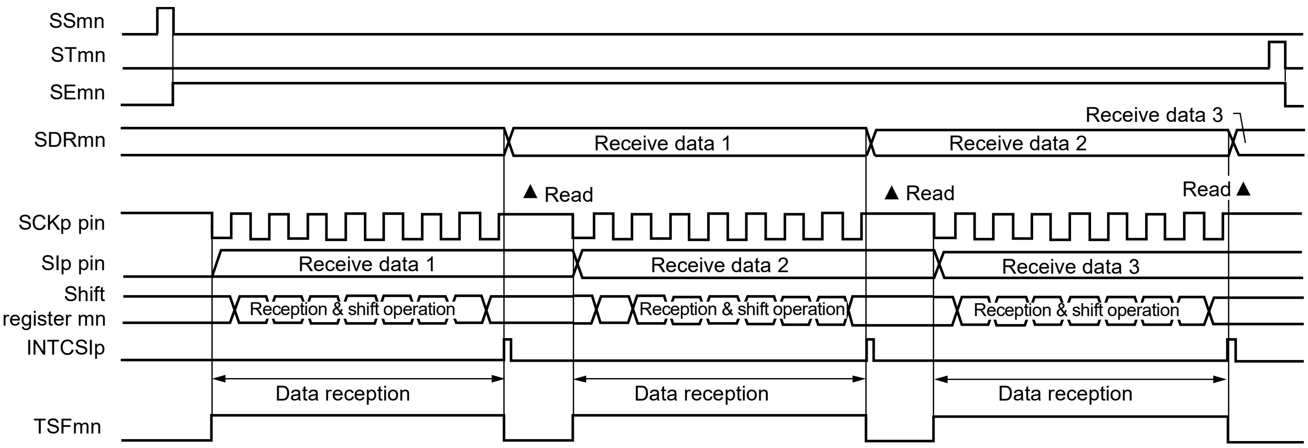
Figure 12-57. Procedure for Resuming Slave Reception



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.

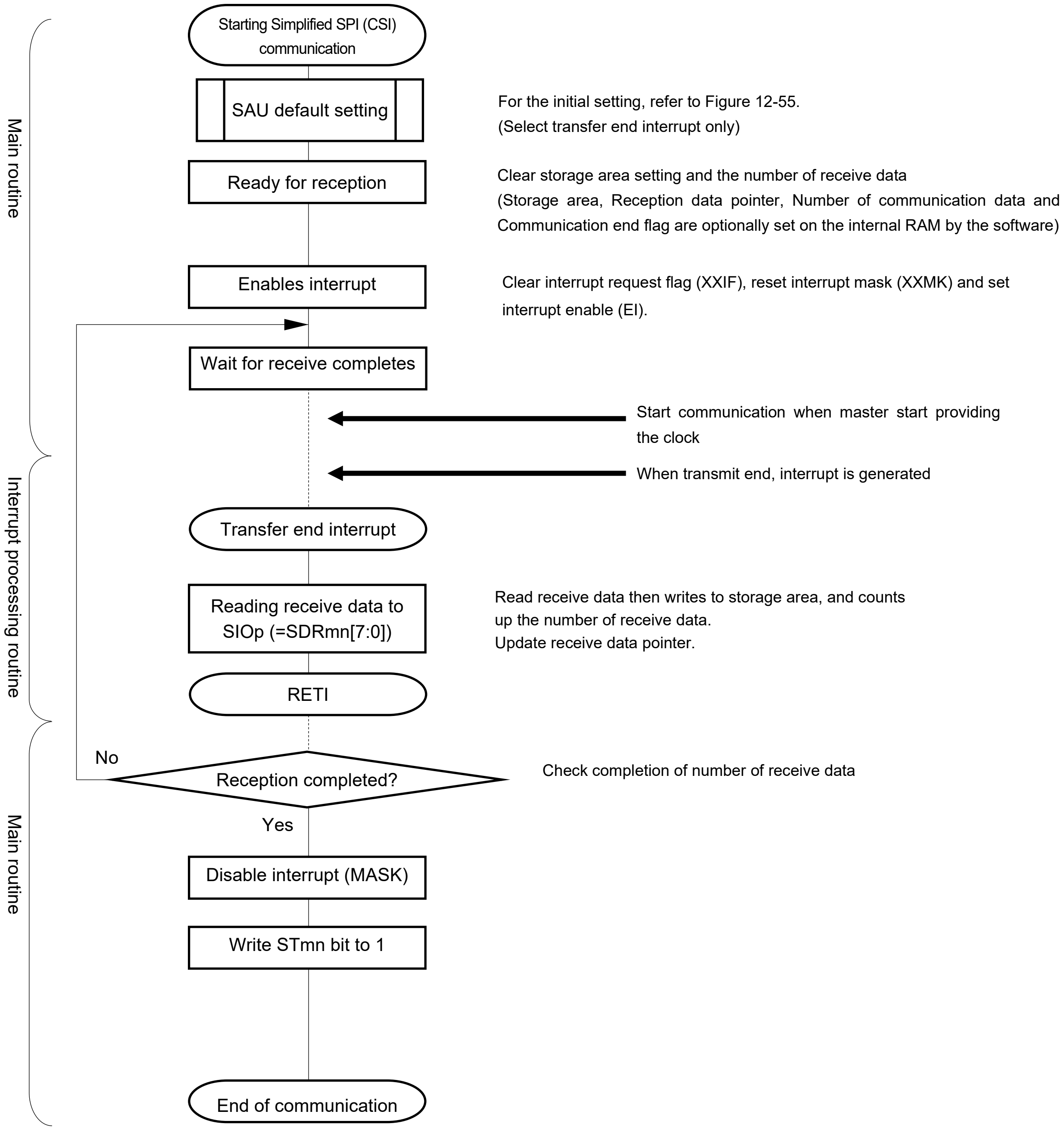
(3) Processing flow (in single-reception mode)

Figure 12-58. Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 12-59. Flowchart of Slave Reception (in Single-Reception Mode)



12.5.6 Slave transmission/reception

Slave transmission/reception is that the R7F0C001G/L, R7F0C002G/L transmits/receives data to/from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0	Channel 1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01
Interrupt	INTCSI00	INTCSI01
	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-reversed ● CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Notes 1. Because the external serial clock input to the SCK00, and SCK01 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 27 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 12-60. Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI01) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	0	MDmn1 0	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **12.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Transmit data setting/receive data register							
									SIOp							

(d) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	1	CKOm1 ×	CKOm0 ×	0	0	0	0	1	1	SOm1 0/1	SOm0 0/1

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

2. ☐ : Setting is fixed in the Simplified SPI (CSI) slave transmission/reception mode,
▨ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-60. Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI01) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	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	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

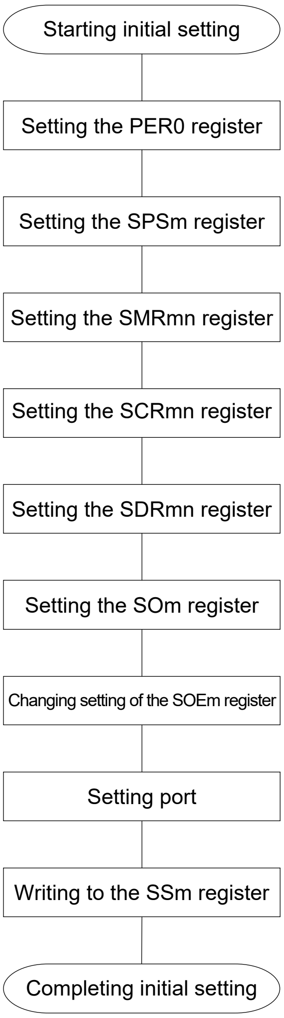
2. ■ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-61. Initial Setting Procedure for Slave Transmission/Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable data output of the target channel.

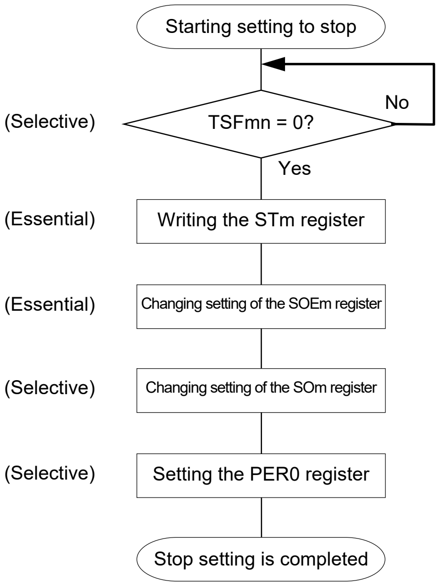
Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Initial setting is completed.
Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Figure 12-62. Procedure for Stopping Slave Transmission/Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write 1 to the STmn bit of the target channel.
(SEmn = 0 : to operation stop status)

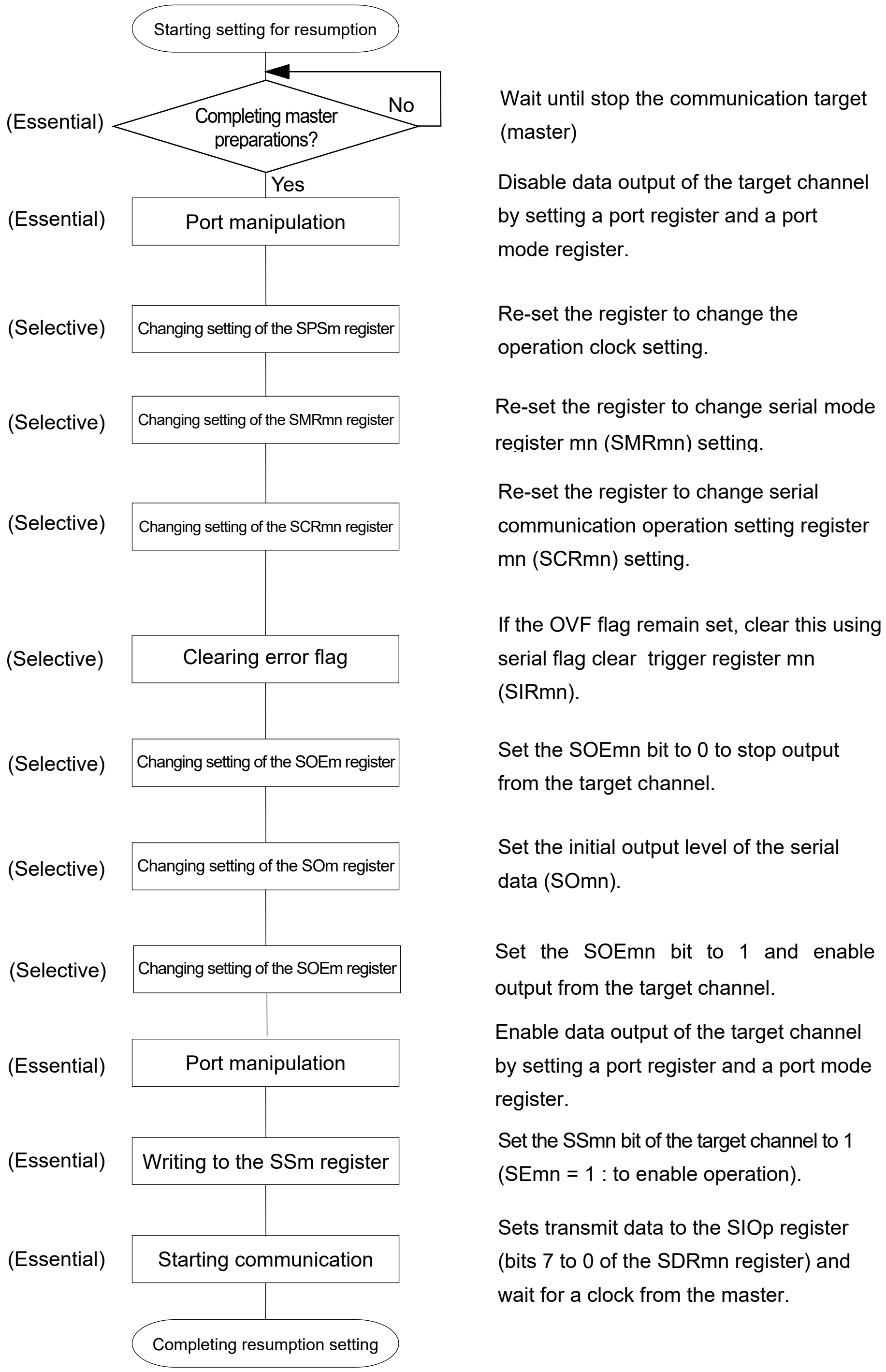
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Figure 12-63. Procedure for Resuming Slave Transmission/Reception

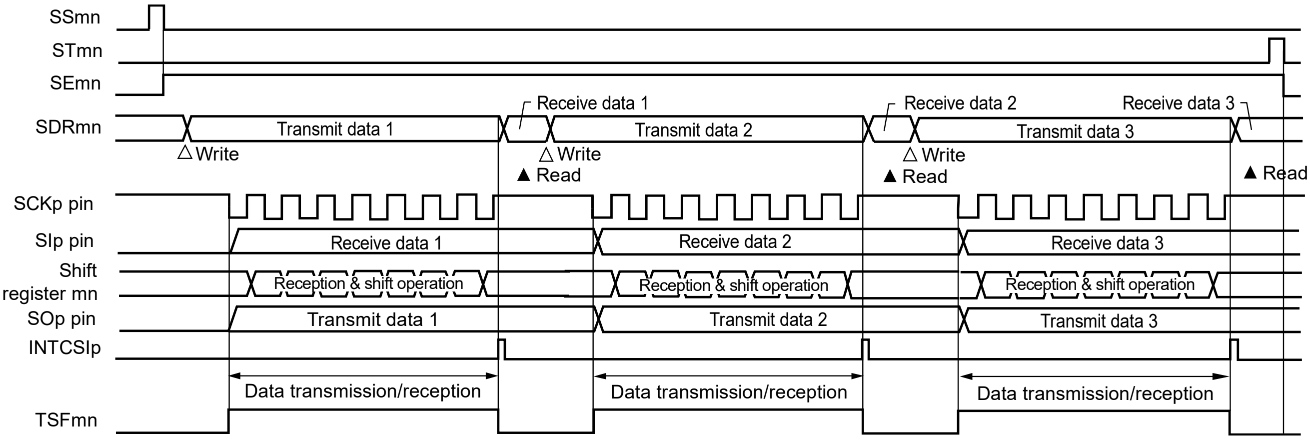


Cautions 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

2. If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

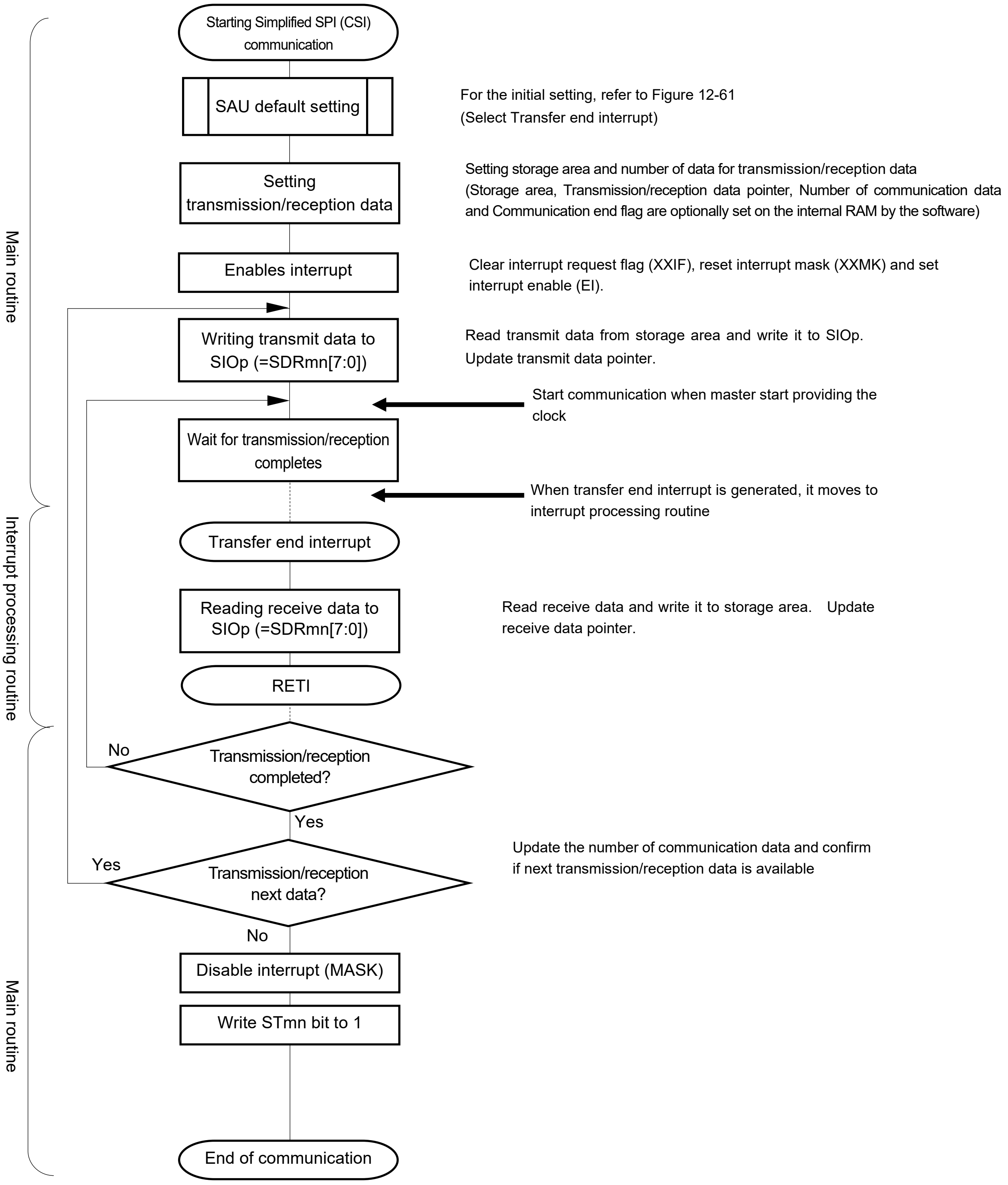
(3) Processing flow (in single-transmission/reception mode)

Figure 12-64. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 12-65. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)



For the initial setting, refer to Figure 12-61 (Select Transfer end interrupt)

Setting storage area and number of data for transmission/reception data (Storage area, Transmission/reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).

Read transmit data from storage area and write it to SIOp. Update transmit data pointer.

Start communication when master start providing the clock

When transfer end interrupt is generated, it moves to interrupt processing routine

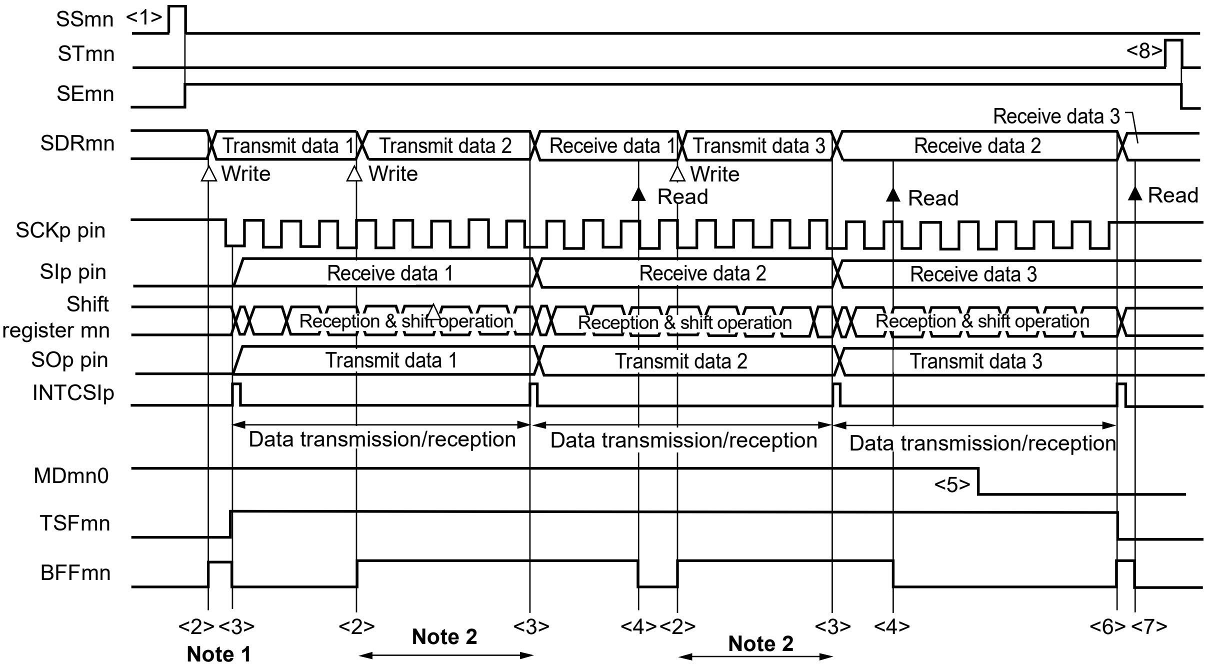
Read receive data and write it to storage area. Update receive data pointer.

Update the number of communication data and confirm if next transmission/reception data is available

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 12-66. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes **1.** If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

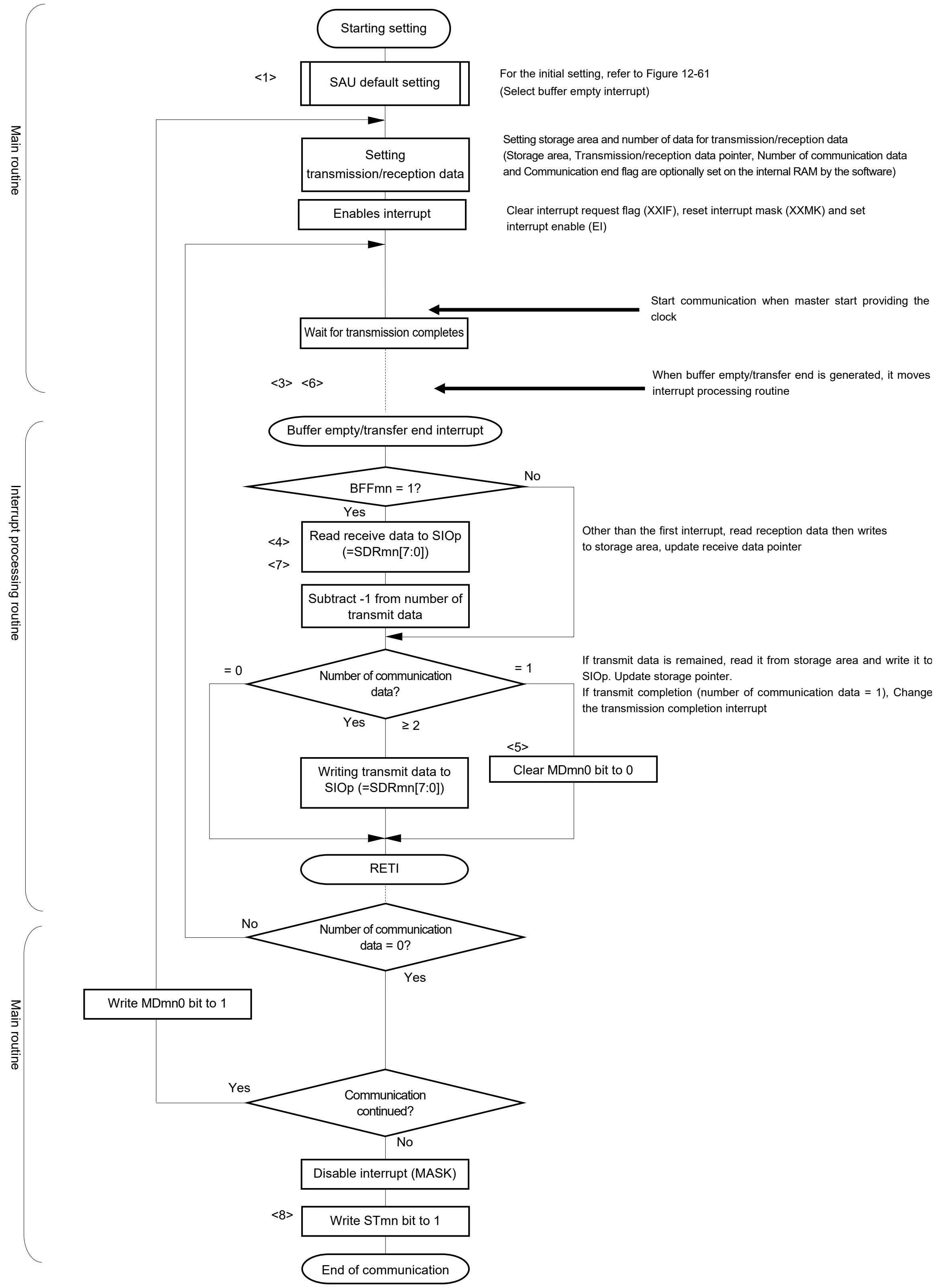
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.**

Remarks **1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-67 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 12-67. 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 12-66 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

12.5.7 SNOOZE mode function

SNOOZE mode makes Simplified SPI (CSI) operate reception by SCKp pin input detection while the STOP mode. Normally Simplified SPI (CSI) stops communication in the STOP mode. But, using the SNOOZE mode makes reception Simplified SPI (CSI) operate unless the CPU operation by detecting SCKp pin input.

Only CSI00 can be set to the SNOOZE mode.

When using the Simplified SPI (CSI) in SNOOZE mode, make the following setting before switching to the STOP mode (see **Figure 12-69 Flowchart of SNOOZE Mode Operation (once startup)** and **Figure 12-71 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.

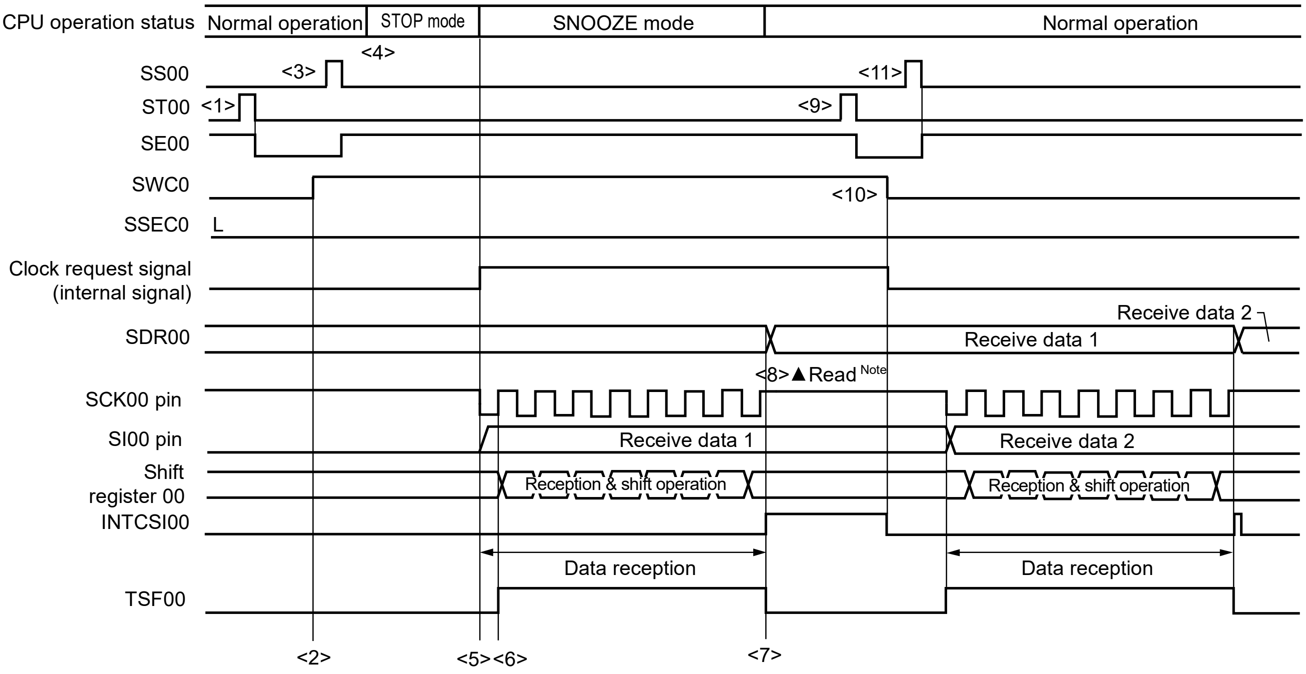
  A CSIp starts reception on detecting input of the serial clock on the SCKp pin.

Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 12-68. 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.

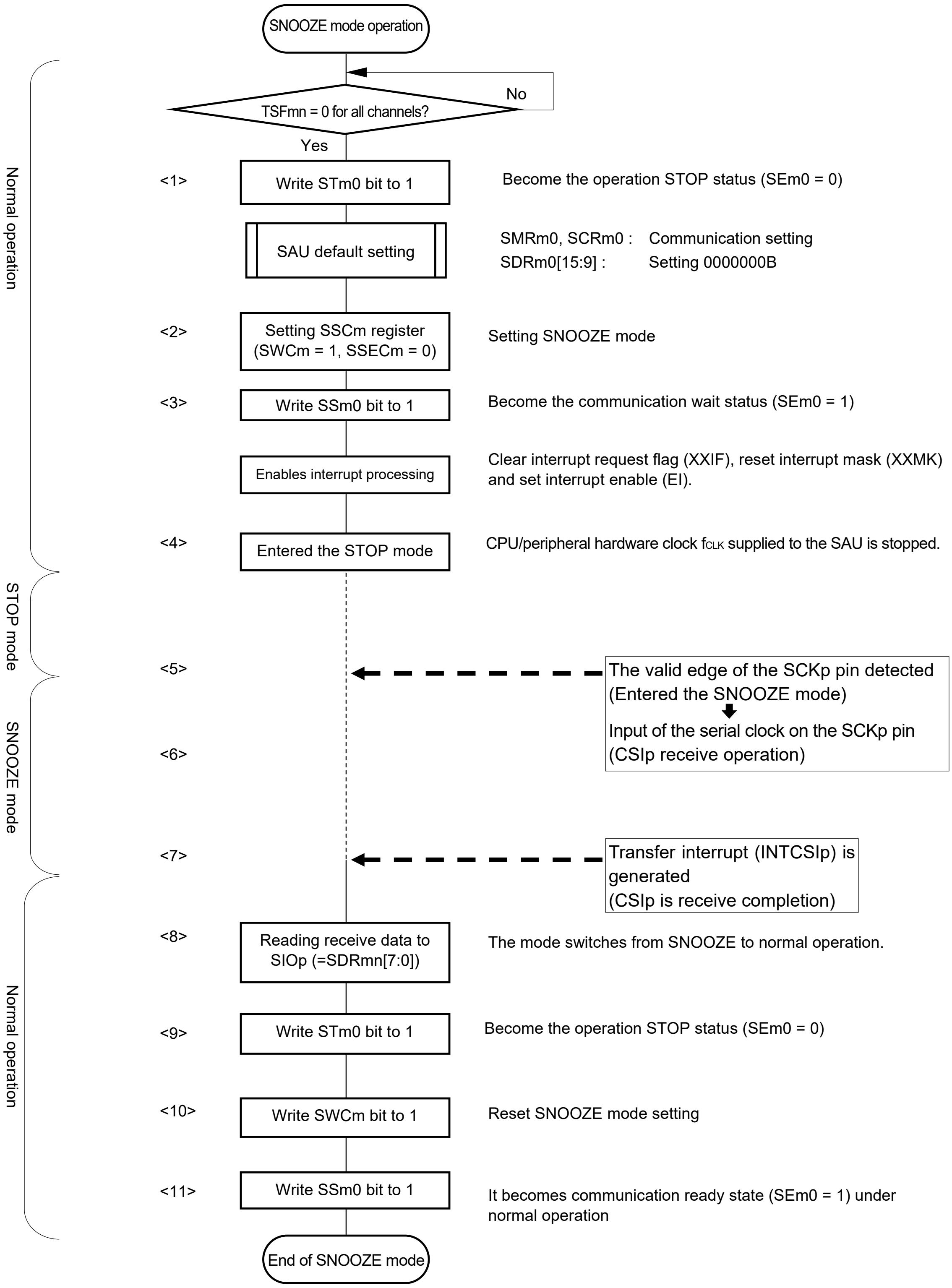
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation). And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

2. When SWCm = 1, the BFFm0 and OVFm0 flags will not change.

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 12-69 Flowchart of SNOOZE Mode Operation (once startup)**.

2. m = 0; p = 00

Figure 12-69. Flowchart of SNOOZE Mode Operation (once startup)

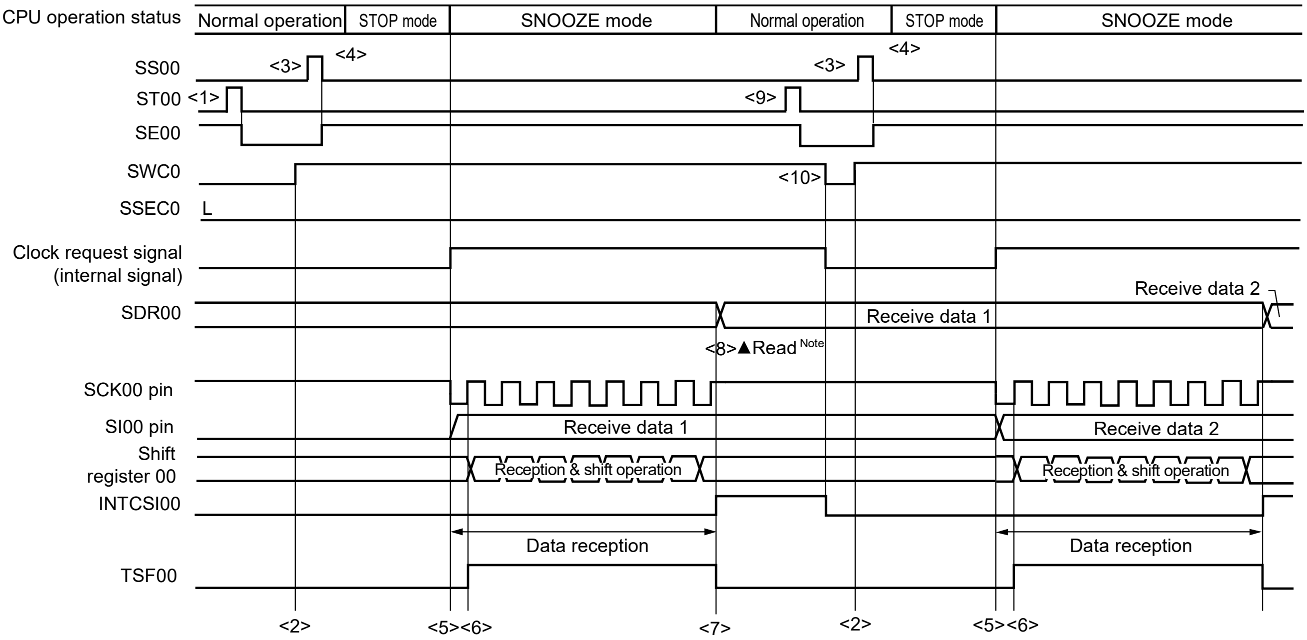


Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 12-68 Timing Chart of SNOOZE Mode Operation (once startup)**.

2. m = 0; p = 00

(2) SNOOZE mode operation (continuous startup)

Figure 12-70. 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.

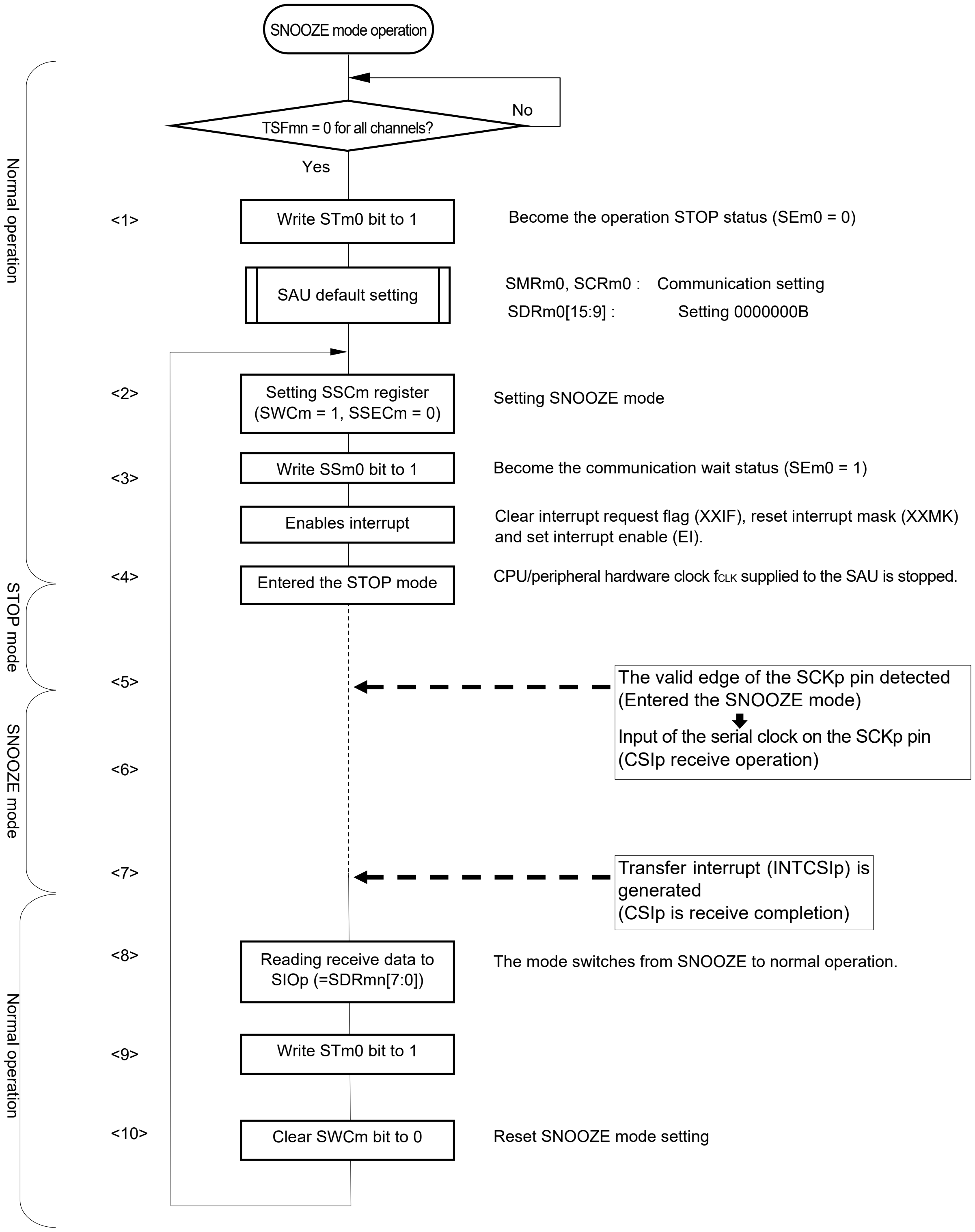
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE release).

2. When SWCm = 1, the BFFm0 and OVFm0 flags will not change.

Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 12-71 Flowchart of SNOOZE Mode Operation (continuous startup)**.

2. m = 0; p = 00

Figure 12-71. Flowchart of SNOOZE Mode Operation (continuous startup)



Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 12-70 Timing Chart of SNOOZE Mode Operation (continuous startup)**.

2. m = 0; p = 00

12.5.8 Calculating transfer clock frequency

The transfer clock frequency for Simplified SPI (CSI00, CSI01) 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 12-2. 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	X	X	X	X	0	0	0	0	f_{CLK}	24 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	12 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	24 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	12 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	732 Hz
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remarks 1. X: Don’t care

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

12.5.9 Procedure for processing errors that occurred during Simplified SPI (CSI00, CSI01) communication

The procedure for processing errors that occurred during Simplified SPI (CSI00, CSI01) communication is described in Figure 12-72.

Figure 12-72. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

12.6 Operation of UART (UART0) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

[Data transmission/reception]

- Data length of 7, 8, or 9 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data (selecting whether to reverse the level)
- Parity bit appending and parity check functions
- Stop bit appending, stop bit check function

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

In addition, UART0 reception supports the SNOOZE mode. When RxD0 pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

UART0 uses channels 0 and 1.

Channel	Used as Simplified SPI (CSI)	Used as UART
0	CSI00	UART0
1	CSI01	

Caution When using serial array unit as UARTs, the channels of both the transmitting side (even-number channel) and the receiving side (odd-number channel) can be used only as UARTs.

UART performs the following two types of communication operations.

- UART transmission	(See **12.6.1**.)
- UART reception	(See **12.6.2**.)

12.6.1 UART transmission

UART transmission is an operation to transmit data from the R7F0C001G/L, R7F0C002G/L to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART0
Target channel	Channel 0
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	7, 8, or 9 bits
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [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	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 Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 27 ELECTRICAL SPECIFICATIONS**).

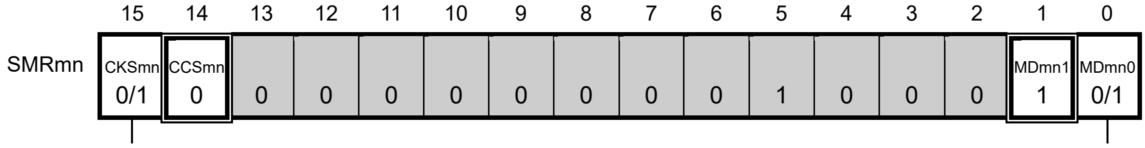
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(1) Register setting

Figure 12-73. Example of Contents of Registers for UART Transmission of UART (UART0) (1/2)

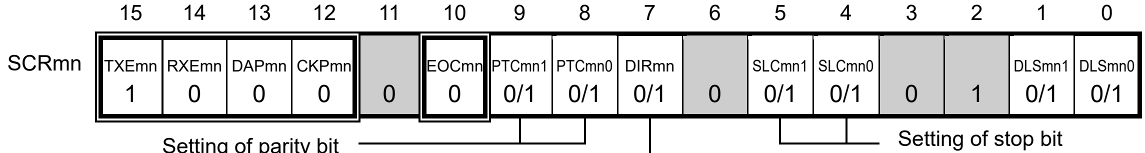
(a) Serial mode register mn (SMRmn)



Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

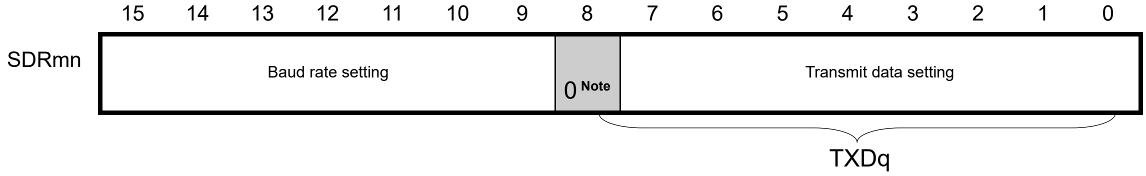


Setting of parity bit
00B: No parity
01B: Appending 0 parity
10B: Appending Even parity
11B: Appending Odd parity

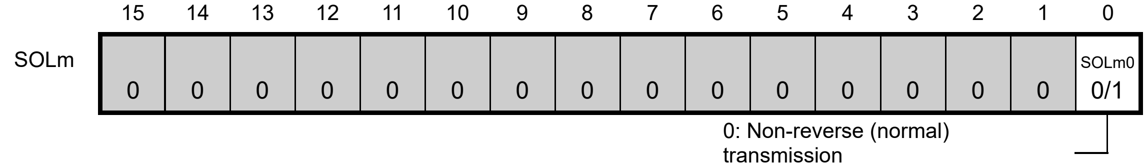
Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of stop bit
01B: Appending 1 bit
10B: Appending 2 bits

(c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



(d) Serial output level register m (SOLm) … Sets only the bits of the target channel.



0: Non-reverse (normal) transmission
1: Reverse transmission

Note 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.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0), mn = 00

2. ☐: Setting is fixed in the UART transmission mode, ▨ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-73. Example of Contents of Registers for UART Transmission of UART (UART0) (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	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 0/1 Note

0: Serial data output value is "0"
1: Serial data output value is "1"

(f) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	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	0	0	SSm1 ×	SSm0 0/1

Note Before transmission is started, be sure to set to 1 when the SOLm0 bit of the target channel is set to 0, and set to 0 when the SOLm0 bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

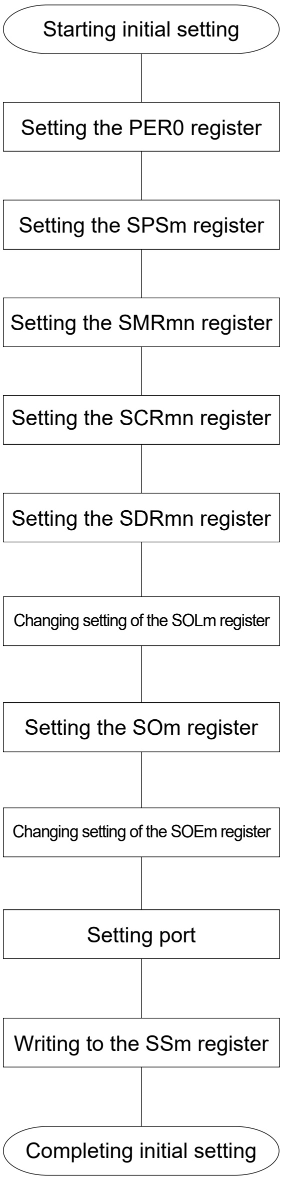
2. ▨ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-74. Initial Setting Procedure for UART Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set an output data level.

Set the initial output level of the serial data (SOmn).

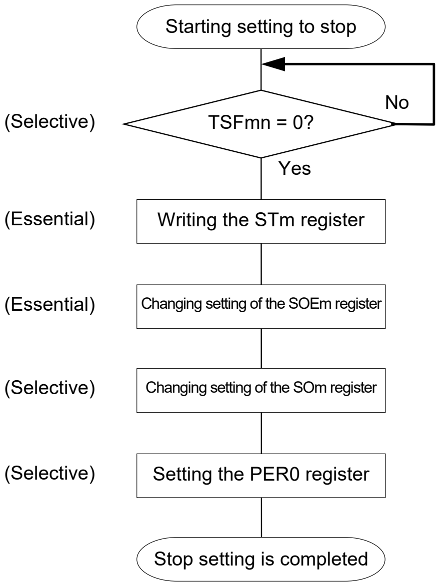
Set the SOEmn bit to 1 and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation).

Initial setting is completed.
Set transmit data to the TXDq register (bits 7 to 0 of the SDRmn register) and start communication.

Figure 12-75 Procedure for Stopping UART Transmission



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write 1 to the STmn bit of the target channel.
(SEmn = 0 : to operation stop status)

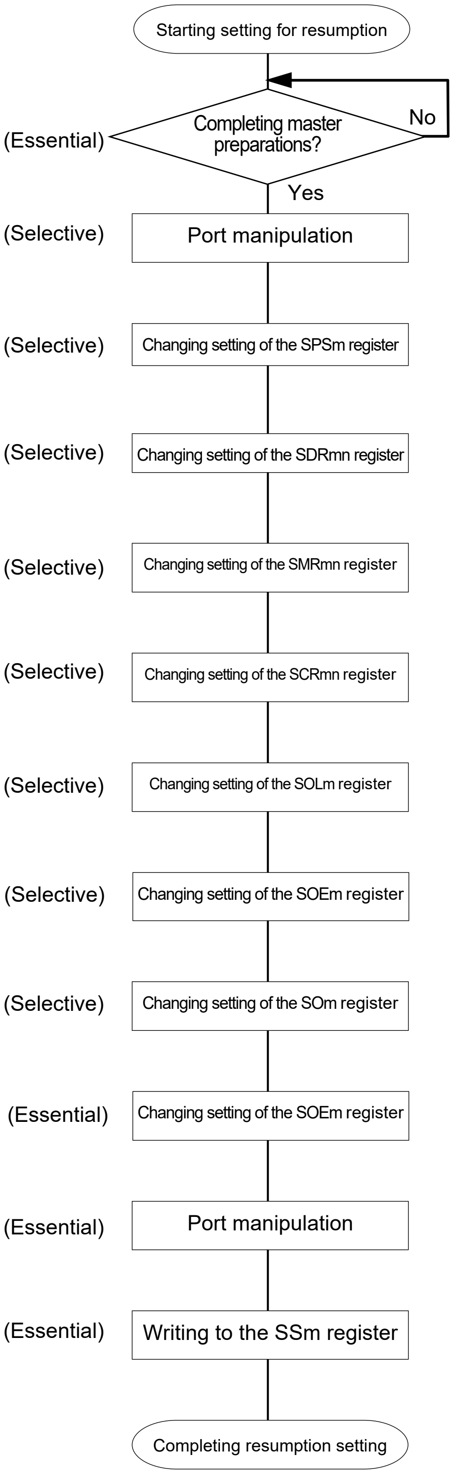
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Figure 12-76. Procedure for Resuming UART Transmission



Wait until stop the communication target or communication operation completed

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change the serial communication operation setting register mn (SCRmn) setting.

Re-set the register to change serial output level register m (SOLm) setting.

Clear the SOEmn bit to 0 and stop output.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output.

Enable data output of the target channel by setting a port register and a port mode register.

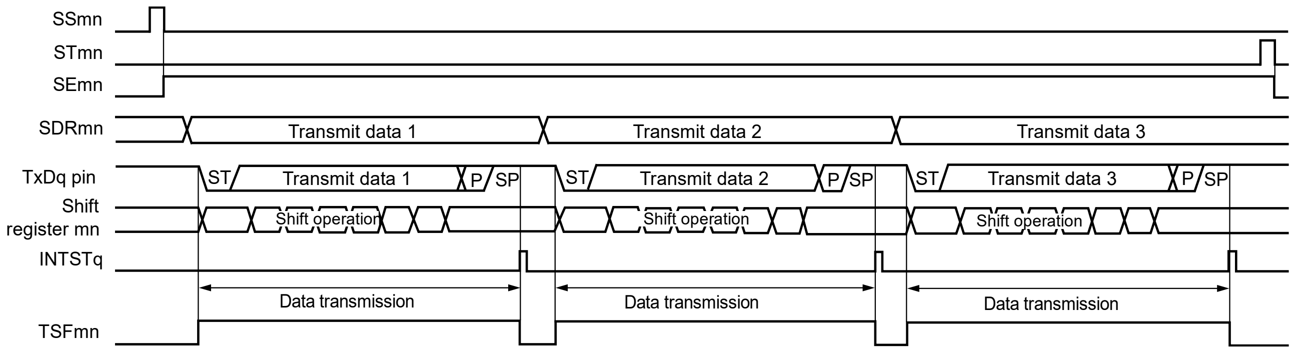
Set the SSmn bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation).

Setting is completed
Set transmit data to the SDRmn[7:0] bits (TXDq register) (8 bits) or the SDRmn[8:0] bits (9 bits) and start communication.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target stops or transmission finishes, and then perform initialization instead of restarting the transmission.

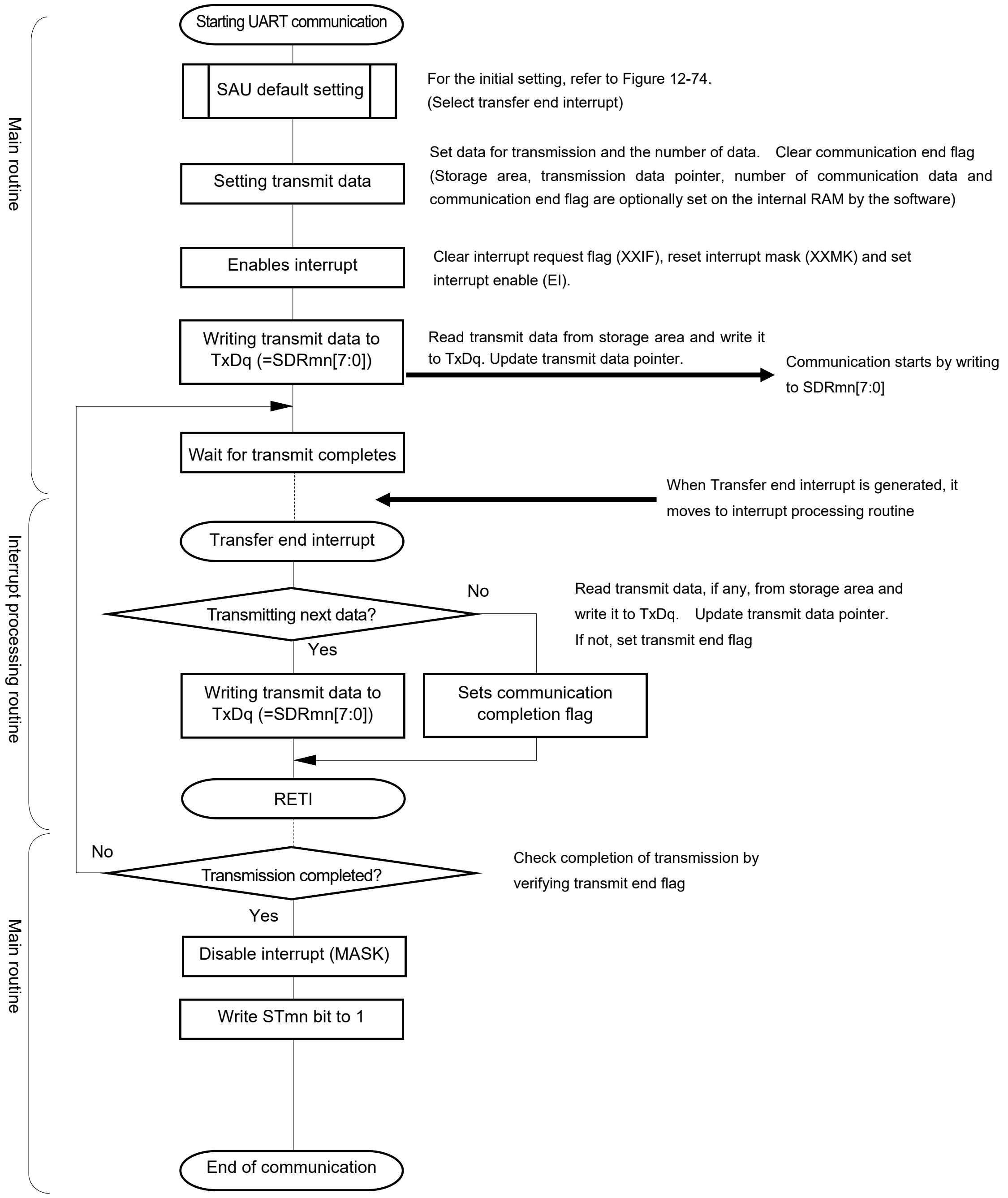
(3) Processing flow (in single-transmission mode)

Figure 12-77. Timing Chart of UART Transmission (in Single-Transmission Mode)



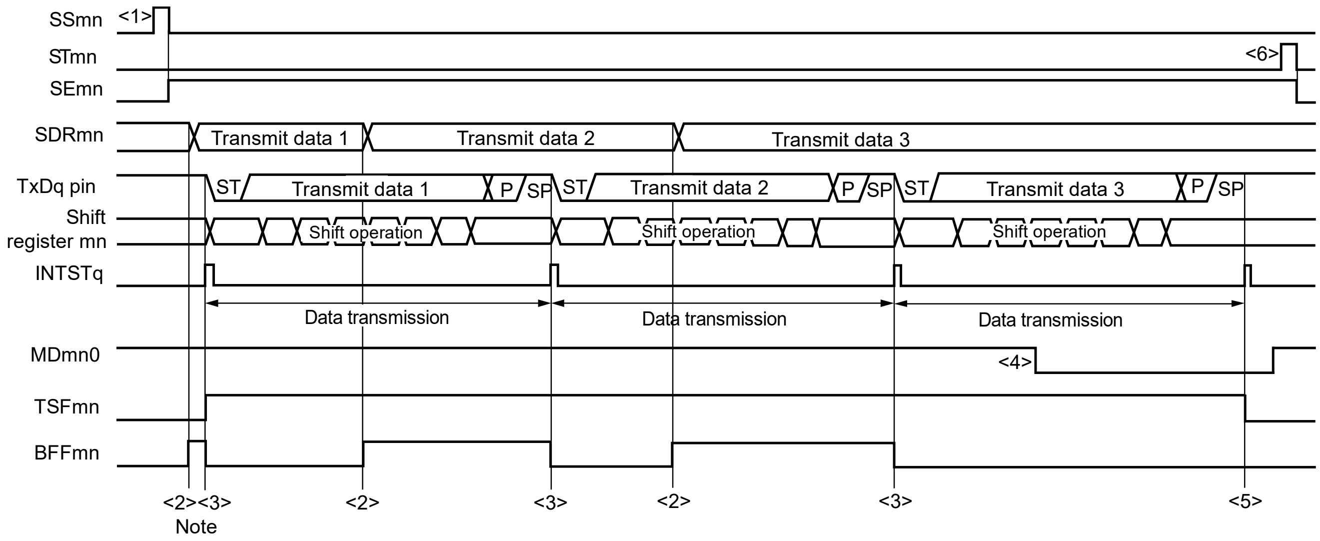
Remark m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0)
mn = 00

Figure 12-78. Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 12-79. Timing Chart of UART Transmission (in Continuous Transmission Mode)

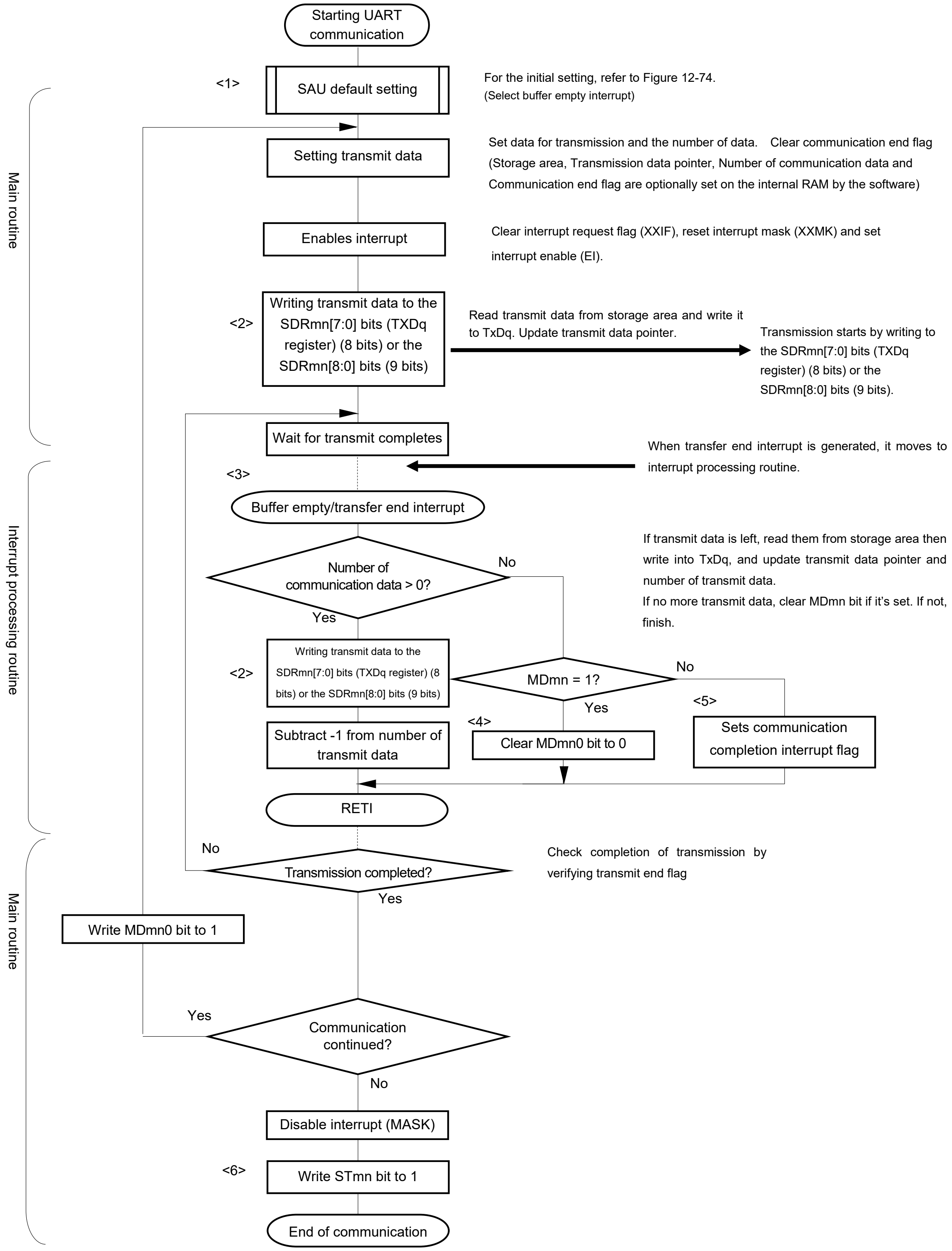


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0)
mn = 00

Figure 12-80. Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-79 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

12.6.2 UART reception

UART reception is an operation wherein the R7F0C001G/L, R7F0C002G/L 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
Target channel	Channel 1
Pins used	RxD0
Interrupt	INTSR0
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error interrupt	INTSRE0
Error detection flag	● Framing error detection flag (FEFmn) ● Parity error detection flag (PEFmn) ● Overrun error detection flag (OVFmn)
Transfer data length	7, 8 or 9 bits
Transfer rate	Max. f_{MCK}/6 [bps] (SDRmn [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	The following selectable ● No parity bit (no parity check) ● No parity judgment (0 parity) ● Appending even parity ● Appending odd parity
Stop bit	Appending 1 bit
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 27 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

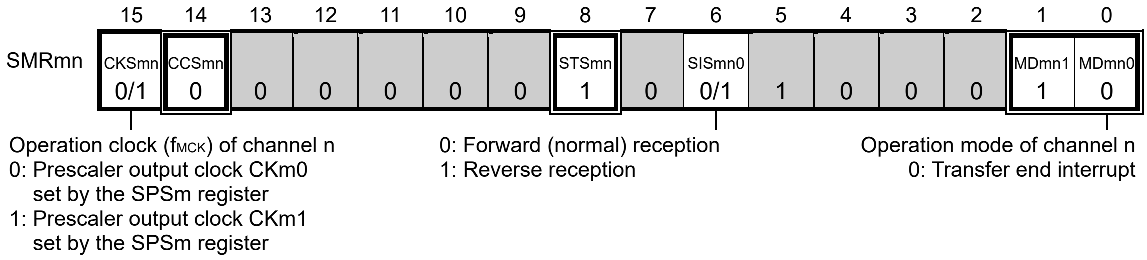
f_{CLK}: System clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01

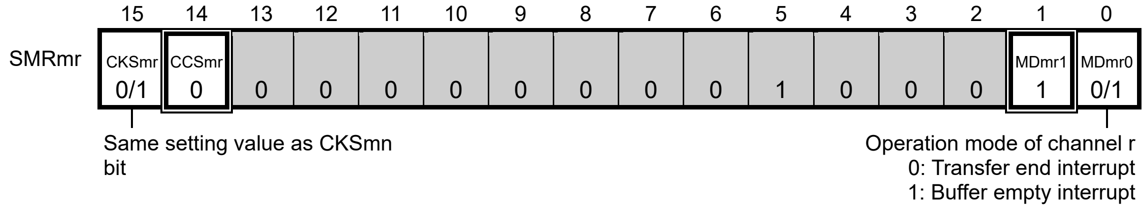
(1) Register setting

Figure 12-81. Example of Contents of Registers for UART Reception of UART (UART0) (1/2)

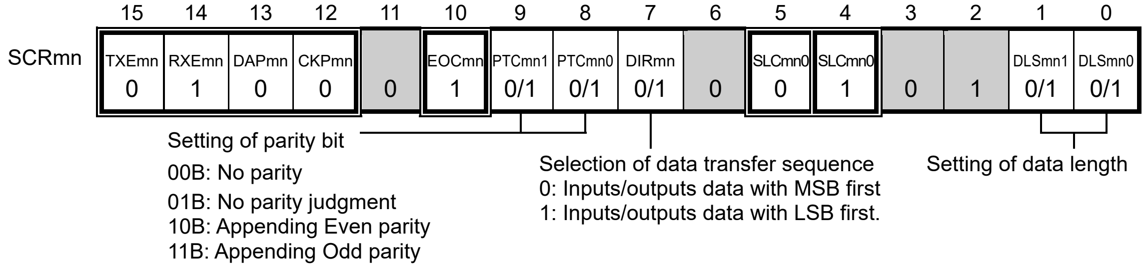
(a) Serial mode register mn (SMRmn)



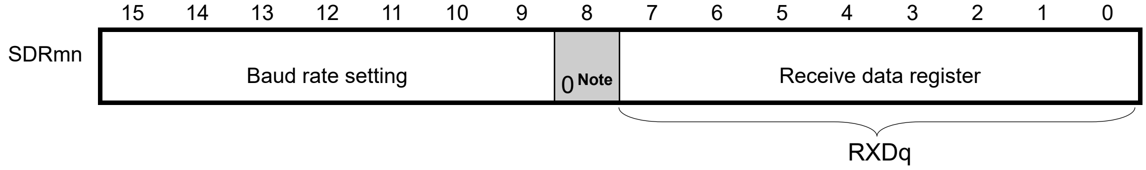
(b) Serial mode register mr (SMRmr)



(c) Serial communication operation setting register mn (SCRmn)



(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



Note When UART performs 9-bit communication, bits 0 to 8 of the SDRm1 register are used as the transmission data specification area.

Caution For the UART reception, be sure to set the SMRmr register of channel r to UART transmission mode that is to be paired with channel n.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01
r: Channel number (r = n – 1), q: UART number (q = 0)

2. ☐: Setting is fixed in the UART reception mode, ▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-81. Example of Contents of Registers for UART Reception of UART (UART0) (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	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	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	0	SOEm1 ×	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	0	0	SSm1 0/1	SSm0 ×

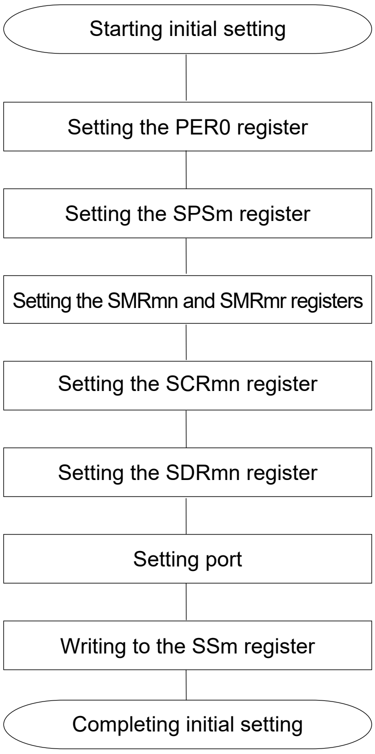
Caution For the UART reception, be sure to set the SMRmr register of channel r to UART Transmission mode that is to be paired with channel n.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01
r: Channel number (r = n – 1), q: UART number (q = 0)

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 12-82. Initial Setting Procedure for UART Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

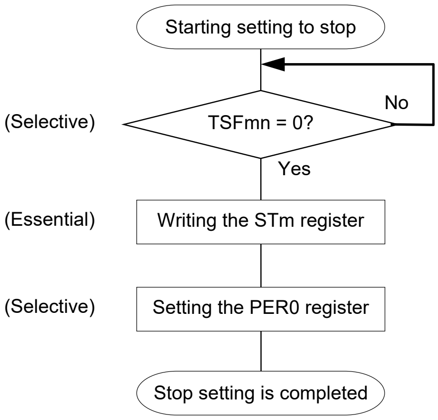
Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Enable data input of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation). Become wait for start bit detection.

Caution Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{CLK} clocks have elapsed.

Figure 12-83. Procedure for Stopping UART Reception



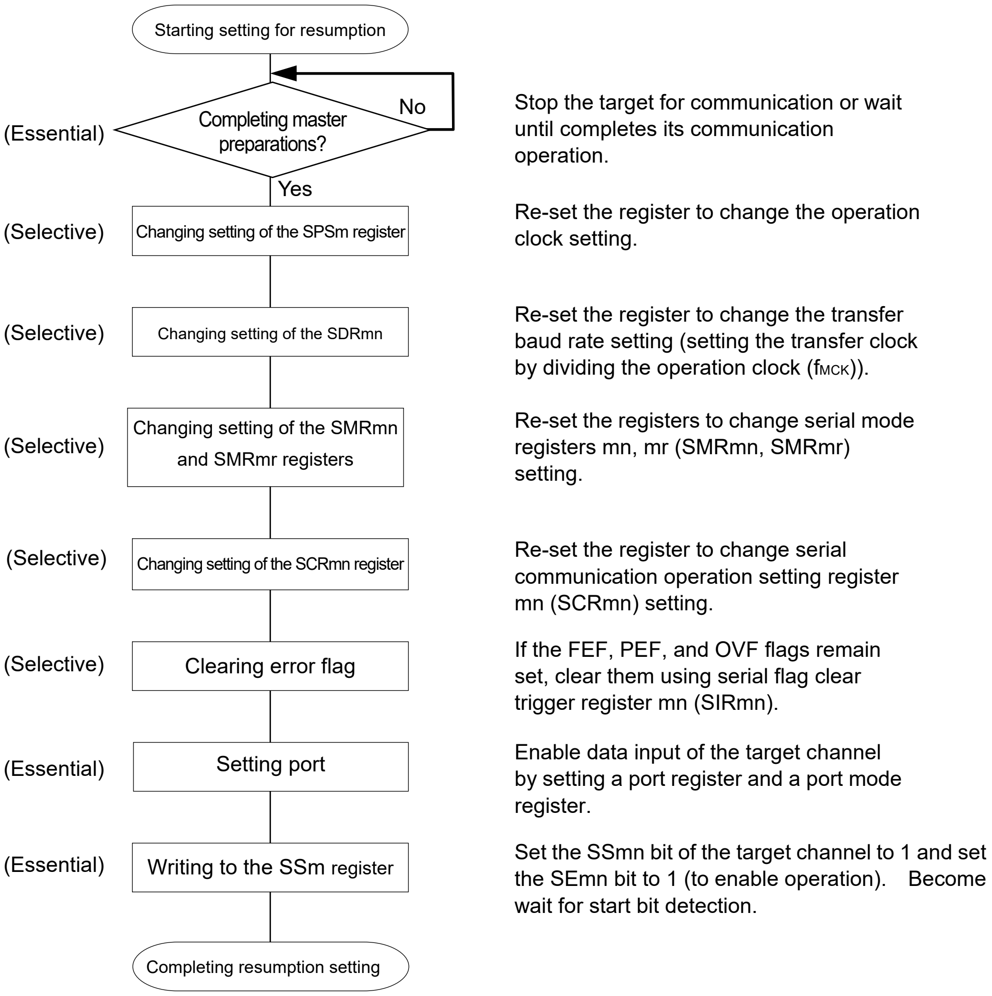
If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write 1 to the STmn bit of the target channel.
(SEmn = 0 : to operation stop status)

To use the STOP mode, reset the serial array unit by stopping the clock supply to it.

The master transmission is stopped.
Go to the next processing.

Figure 12-84. Procedure for Resuming UART Reception

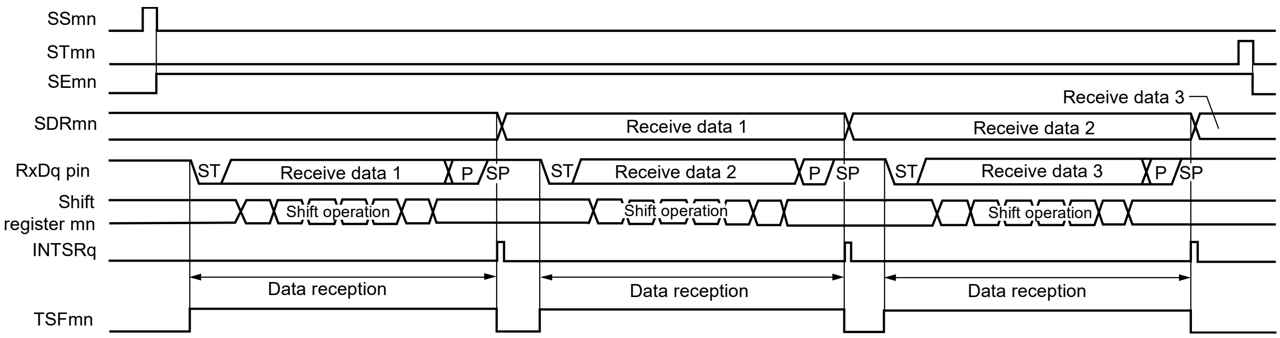


Caution **After is set RXEmn bit to 1 of SCRmn register, set the SSmn = 1 from an interval of at least four clocks of f_{MCK}.**

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target stops or transmission finishes, and then perform initialization instead of restarting the transmission.

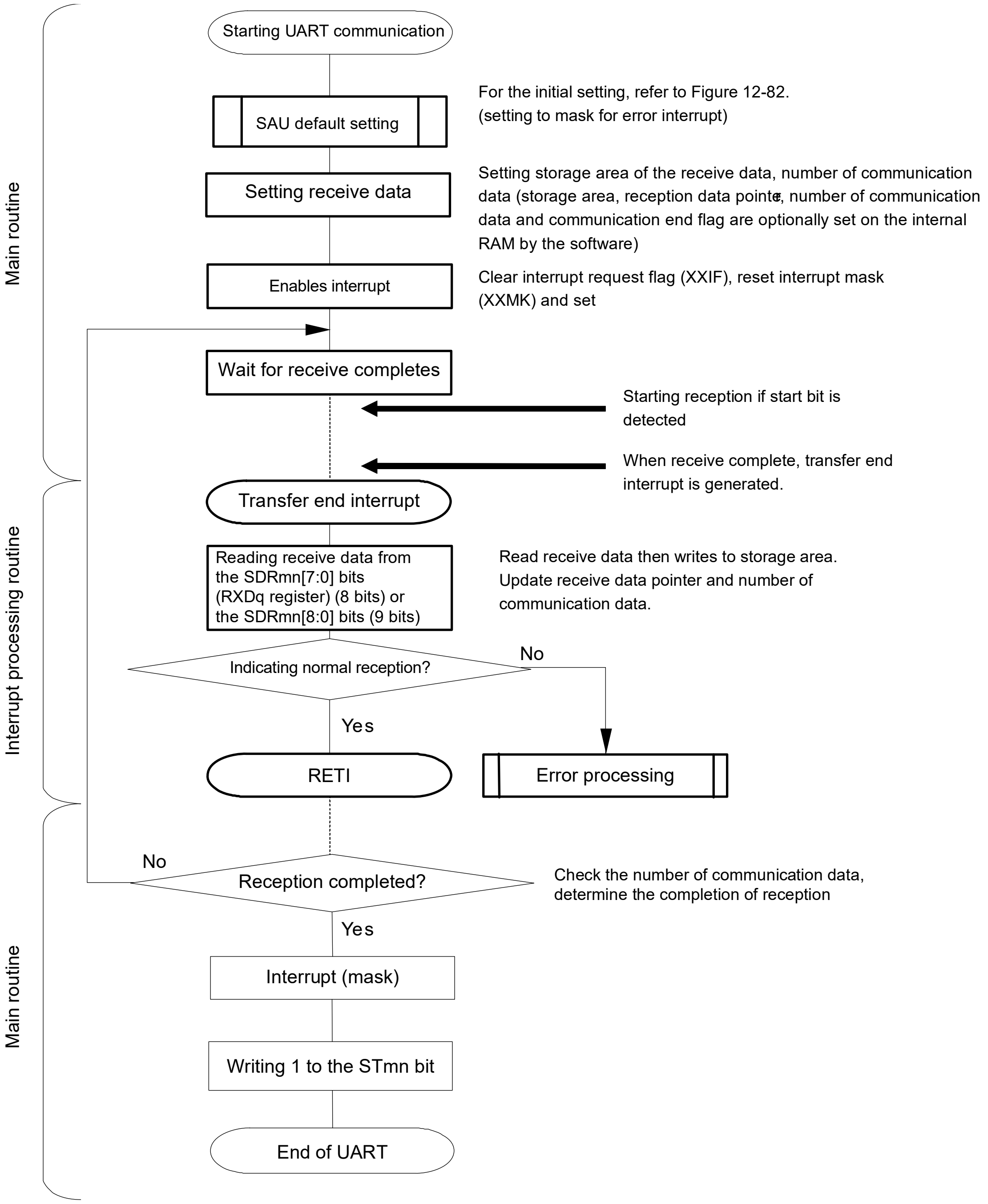
(3) Processing flow

Figure 12-85. Timing Chart of UART Reception



Remark m: Unit number (m = 0), n: Channel number (n = 1), mn = 01
r: Channel number (r = n – 1), q: UART number (q = 0)

Figure 12-86. Flowchart of UART Reception



12.6.3 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 upon detection of the RxDq pin input.

Only UART0 can be set to the SNOOZE mode.

When using UARTq in the SNOOZE mode, make the following settings before entering the STOP mode. (See **Figure 12-89 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)** and **Figure 12-91 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 12-3.
- Set the EOCmn and SSECmn bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has completed, set the SSm1 bit of serial channel start register m (SSm) to 1.
- 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.

Cautions 1. The SNOOZE mode can only be used when the high-speed on-chip oscillator clock (f_{IH}) is selected for f_{CLK}.

2. The transfer rate in the SNOOZE mode is only 4800 bps.

3. When SWCm = 1, UARTq can be used only when the reception operation is started in the STOP mode. When used simultaneously with another SNOOZE mode function or interrupt, if the reception operation is started in a state other than the STOP mode, such as those given below, data may not be received correctly and a framing error or parity error may be generated.

- **When after the SWCm bit has been set to 1, the reception operation is started before the STOP mode is entered**
- **When the reception operation is started while another function is in the SNOOZE mode**
- **When after returning from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWCm bit is returned to 0**

4. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFmn, FEFmn, or OVFmn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFmn, FEFmn, or OVFmn flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

5. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxDq signal. Note, however, that transfer through the UART channel may not start and the CPU may remain in the SNOOZE mode if an input pulse on the RxDq pin is too short to be detected as a start bit. In such cases, data may not be received correctly, and this may lead to a framing error or parity error in the next UART transfer.

Table 12-3. Baud Rate Setting for UART Reception in SNOOZE Mode

High-speed On-chip Oscillator (f_{IH})	Baud Rate for UART Reception in SNOOZE Mode			
	Baud Rate of 4800 bps			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Maximum Permissible Value	Minimum Permissible Value
32 MHz ± 1.0% Note	$f_{CLK}/2^5$	105	2.27%	–1.53%
24 MHz ± 1.0% Note	$f_{CLK}/2^5$	79	1.60%	–2.18%
16 MHz ± 1.0% Note	$f_{CLK}/2^4$	105	2.27%	–1.53%
12 MHz ± 1.0% Note	$f_{CLK}/2^4$	79	1.60%	–2.19%
8 MHz ± 1.0% Note	$f_{CLK}/2^3$	105	2.27%	–1.53%
6 MHz ± 1.0% Note	$f_{CLK}/2^3$	79	1.60%	–2.19%
4 MHz ± 1.0% Note	$f_{CLK}/2^2$	105	2.27%	–1.53%
3 MHz ± 1.0% Note	$f_{CLK}/2^2$	79	1.60%	–2.19%
2 MHz ± 1.0% Note	$f_{CLK}/2$	105	2.27%	–1.54%
1 MHz ± 1.0% Note	f_{CLK}	105	2.27%	–1.57%

Note When the accuracy of the clock frequency of the high-speed on-chip oscillator is ±1.5% or ±2.0%, the permissible range becomes smaller as shown below.

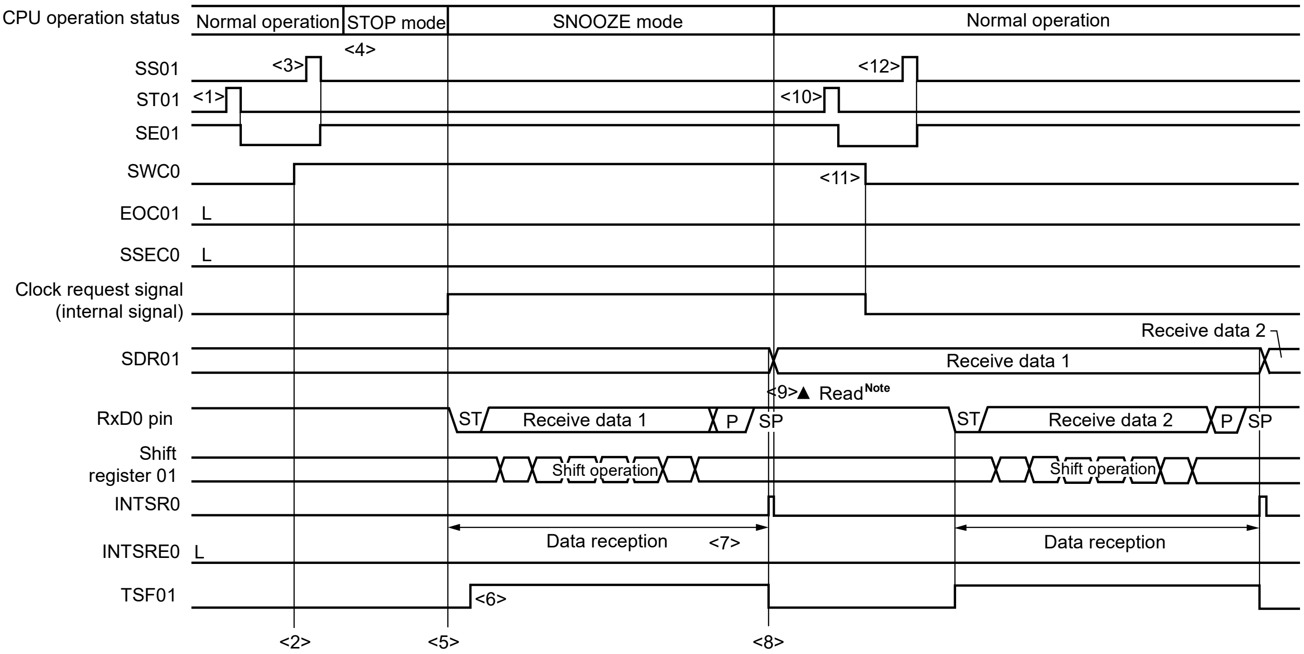
- In the case of f_{IH} ± 1.5%, perform (Maximum permissible value **–** 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of f_{IH} ± 2.0%, perform (Maximum permissible value **–** 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

Remark The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception. The baud rate on the transmitting side should be set to fall inside this range.

(1) SNOOZE mode operation (EOCm1 = 0, SSECm = 0/1)

Because of the setting of EOCm1 = 0, even though a communication error occurs, an error interrupt (INTSREq) is not generated, regardless of the setting of the SSECm bit. A transfer end interrupt (INTSRq) will be generated.

Figure 12-87. Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm = 1.

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit and stop the operation).
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

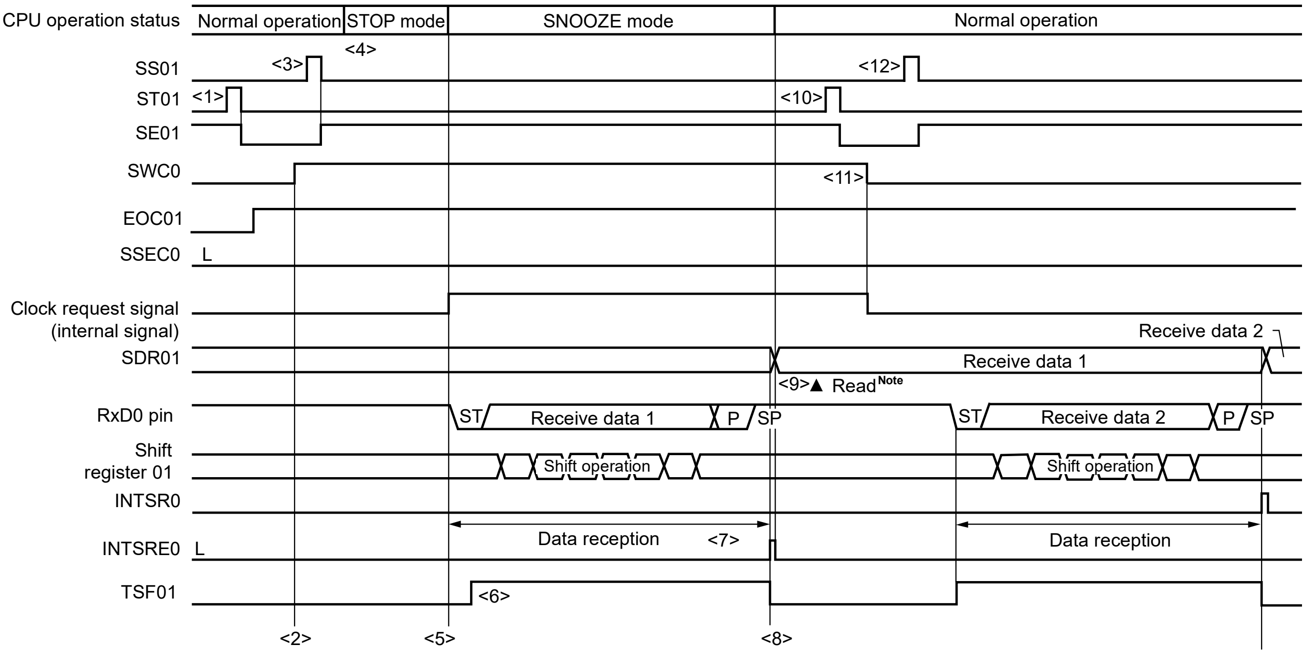
Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 12-89 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0

(2) SNOOZE mode operation (EOCm1 = 1, SSECm = 0: Error interrupt (INTSREq) generation is enabled)

Because EOCm1 = 1 and SSECm = 0, an error interrupt (INTSREq) is generated when a communication error occurs.

Figure 12-88. Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



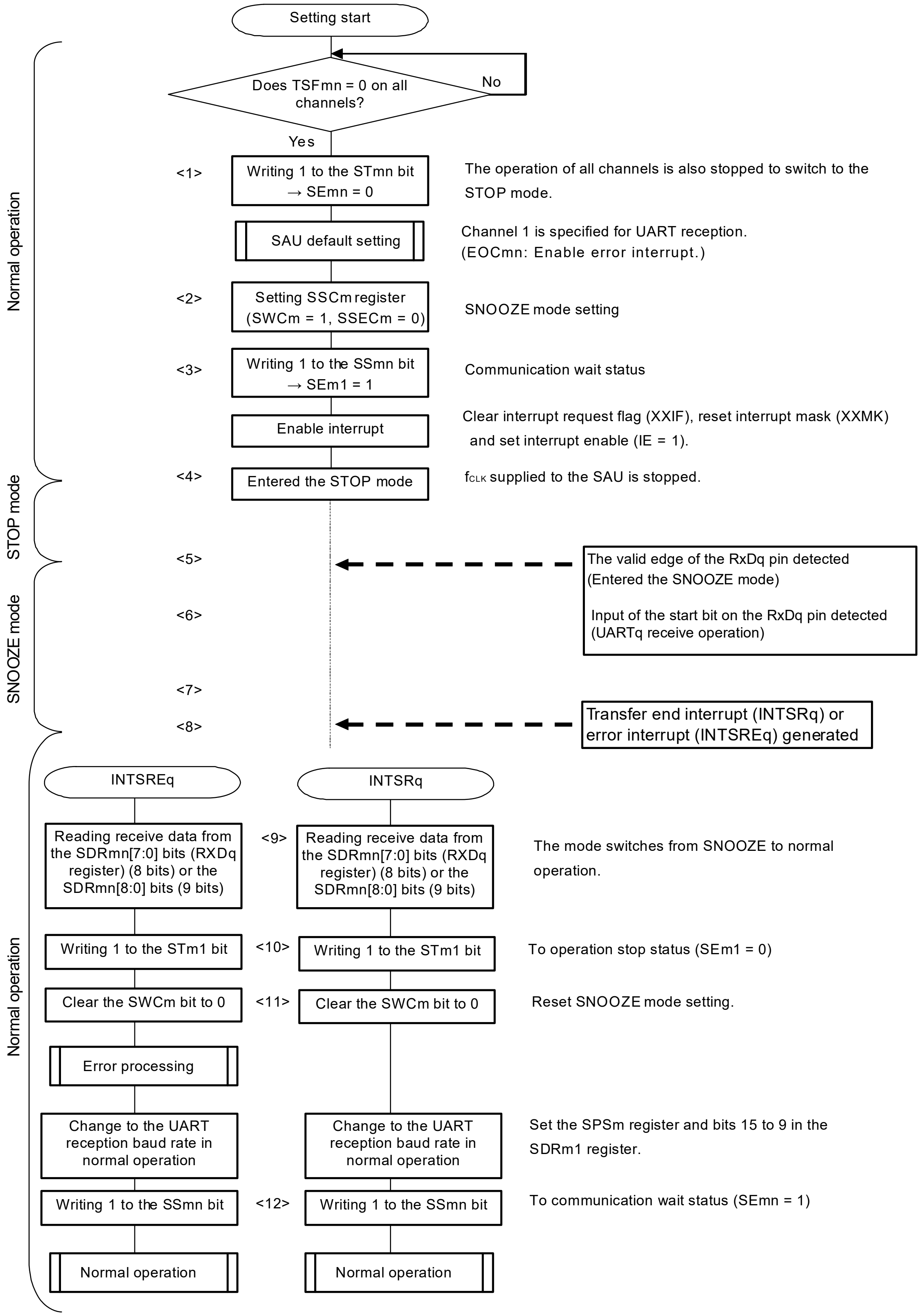
Note Read the received data when SWCm = 1.

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit and stop the operation).
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 12-89 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0

Figure 12-89. Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)



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

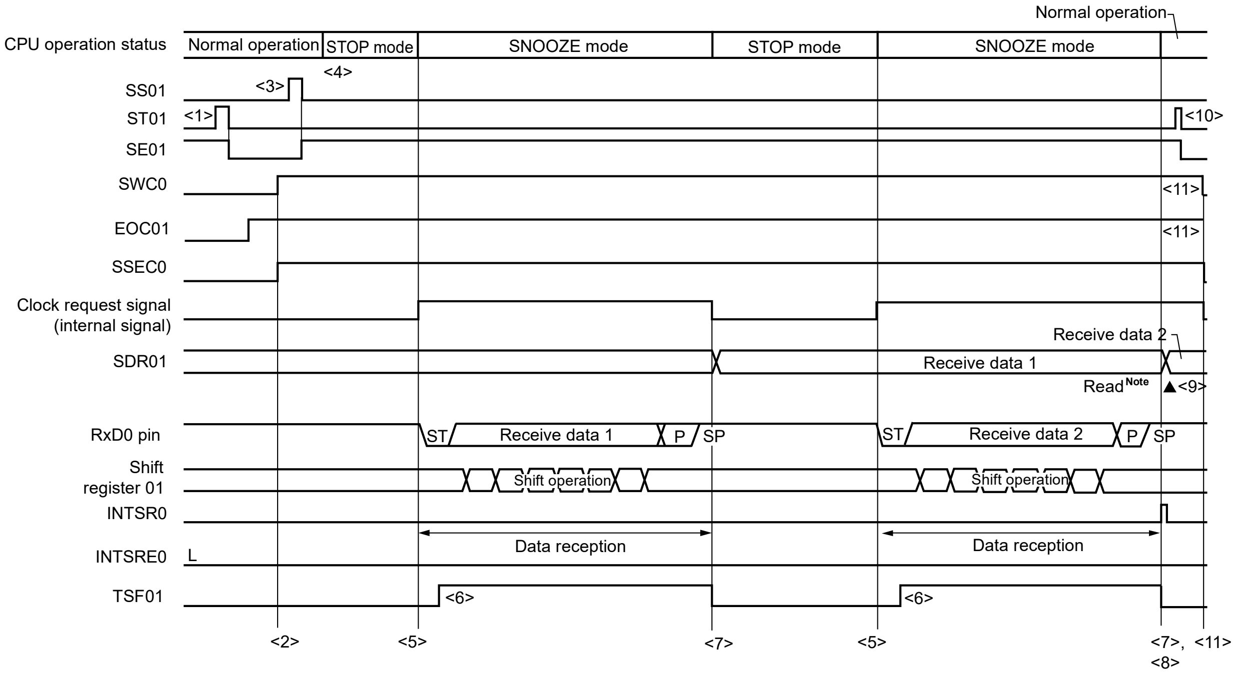
Remarks 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 12-87 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)** and **Figure 12-88 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)**.

2. m = 0; q = 0

(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 12-90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Read the received data when SWCm = 1.

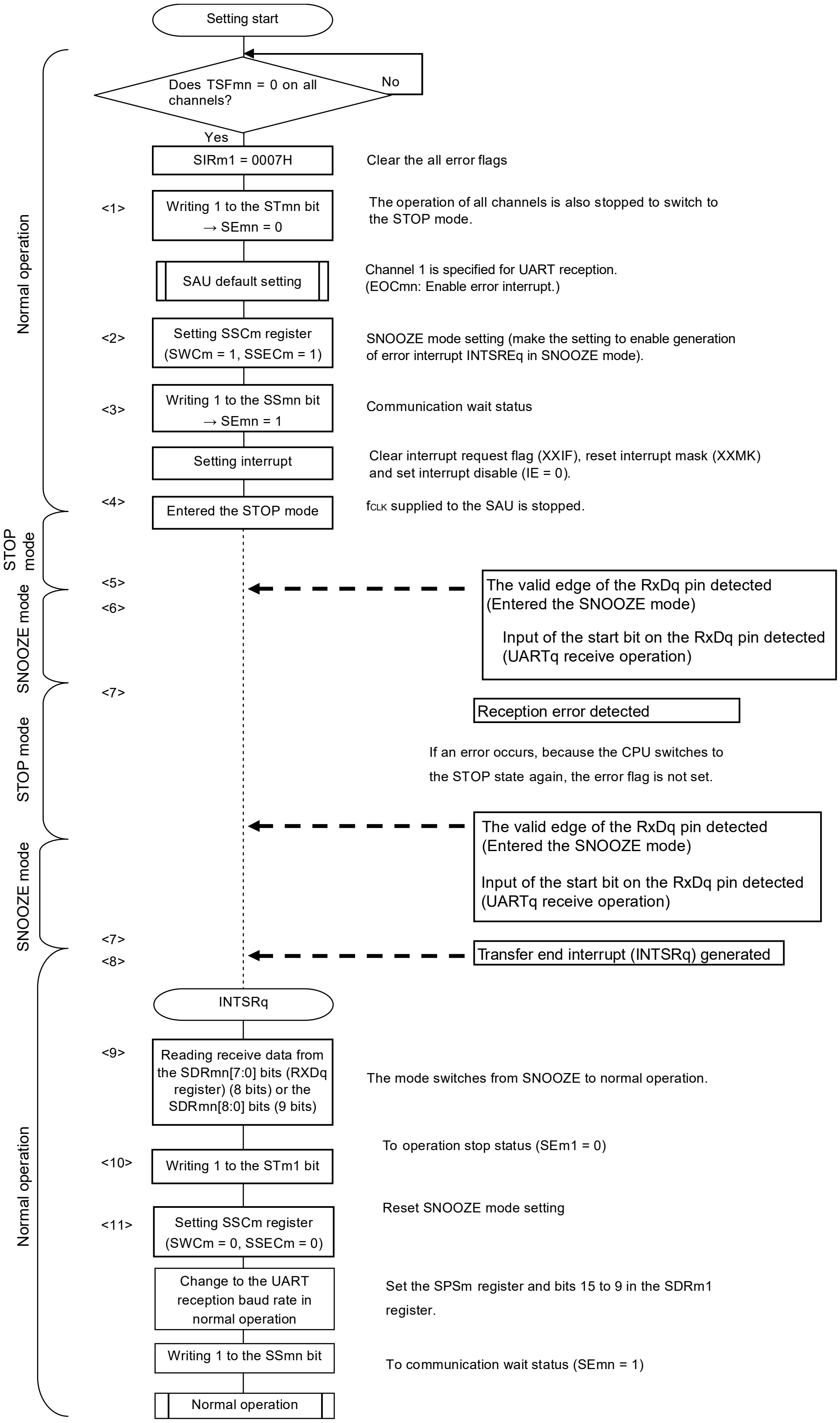
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit and stop the operation).
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

2. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 12-91 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

2. m = 0; q = 0

Figure 12-91. 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).

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 12-90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

2. m = 0; q = 0

12.6.4 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

Remarks 1. When UART is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B) and therefore is 2 to 127.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

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 12-4. 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	X	X	X	X	0	0	0	0	f_{CLK}	24 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	12 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^{2}$	6 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^{3}$	3 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^{4}$	1.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^{5}$	750 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^{6}$	375 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^{7}$	187.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^{8}$	93.8 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^{9}$	46.9 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	24 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	12 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^{2}$	6 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^{3}$	3 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^{4}$	1.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^{5}$	750 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^{6}$	375 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^{7}$	187.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^{8}$	93.8 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^{9}$	46.9 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	732 Hz
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remarks 1. X: Don’t care

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(2) Baud rate error during transmission

The baud rate error of UART (UART0) 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), n: Channel number (n = 0), mn = 00

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0) 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.

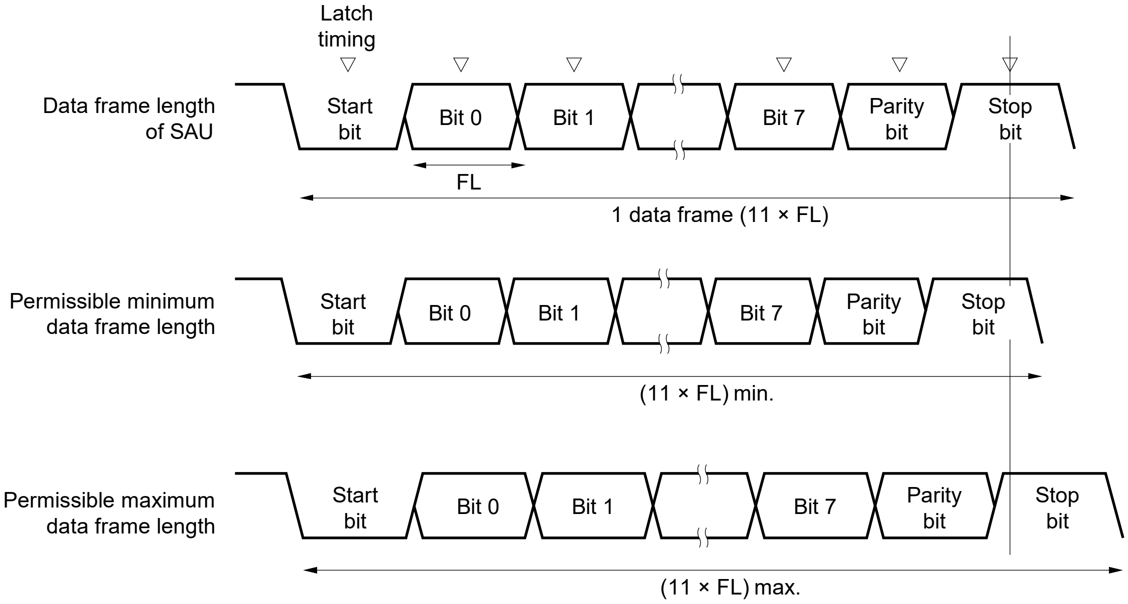
$$(\text{Maximum receivable baud rate}) = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times \text{Brate}$$

$$(\text{Minimum receivable baud rate}) = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times \text{Brate}$$

Brate: Calculated baud rate value at the reception side (See **12.6.4 (1) Baud rate calculation expression**.)
k: SDRmn[15:9] + 1
Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0), n: Channel number (n = 1), mn = 01

Figure 12-92. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 12-92, 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.

12.6.5 Procedure for processing errors that occurred during UART (UART0) communication

The procedure for processing errors that occurred during UART (UART0) communication is described in Figures 12-93 and 12-94.

Figure 12-93. 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 12-94. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn).	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn).	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1.	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operating.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SSmn bit of serial channel start register m (SSm) to 1.	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

CHAPTER 13 LCD CONTROLLER/DRIVER

The number of LCD display function pins of the R7F0C001G/L, R7F0C002G/L differs depending on the product. The following table shows the number of pins of each product.

Table 13-1. Number of LCD Display Function Pins of Each Product

Item \ Part No.		R7F0C001G/L, R7F0C002G/L															
		48 pins								64 pins							
Number of LCD output pins		Segment signal outputs: 26 (22)[Note] Common signal outputs: 8								Segment signal outputs: 39 (35)[Note] Common signal outputs: 8							
Correspondence between multiplexed segment signal output pin functions and I/O port pin functions	I/O port \ x	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	P1x	SEG 6	SEG 5	SEG 4	SEG 32	SEG 31	SEG 30	SEG 29	SEG 28	SEG 6	SEG 5	SEG 4	SEG 32	SEG 31	SEG 30	SEG 29	SEG 28
	P3x	–	–	–	–	–	SEG 17	SEG 18	SEG 19	–	–	–	–	–	SEG 17	SEG 18	SEG 19
	P4x	–	–	–	–	–	–	SEG 24	–	–	–	–	–	SEG 22	SEG 23	SEG 24	–
	P5x	–	–	–	–	–	–	–	SEG 7	–	–	–	SEG 11	SEG 10	SEG 9	SEG 8	SEG 7
	P6x	–	–	–	–	–	–	SEG 20	SEG 21	–	–	–	–	–	–	SEG 20	SEG 21
	P7x	–	–	–	–	–	–	–	SEG 16	–	–	–	SEG 12	SEG 13	SEG 14	SEG 15	SEG 16
	P12x	–	–	–	–	–	–	–	SEG 25	–	–	–	–	–	–	–	SEG 25
	P14x	–	–	–	SEG 35	SEG 34	SEG 33	SEG 26	SEG 27	SEG 38	SEG 37	SEG 36	SEG 35	SEG 34	SEG 33	SEG 26	SEG 27
Correspondence between multiplexed COM signal output pin functions and I/O port pin functions		–								–							
Correspondence between multiplexed COM signal output pin functions and other LCD display pin functions		SEG0								SEG0							
		SEG1								SEG1							
		SEG2								SEG2							
		SEG3								SEG3							

Note ( ) indicates the number of signal output pins when 8 com is used.

13.1 Functions of LCD Controller/Driver

The functions of the LCD controller/driver in the R7F0C001G/L, R7F0C002G/L microcontrollers are as follows.

(1) Waveform A or B 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) The LCD can be made to blink **Note**

Note Selecting f_{IL} as the LCD source clock (f_{LCD}) is prohibited.

Table 13-2 lists the maximum number of pixels that can be displayed in each display mode.

Table 13-2. Maximum Number of Pixels

(a) 48-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	26 (26 segment signals, 1 common signal)
		1/2	2	52 (26 segment signals, 2 common signals)
			3	78 (26 segment signals, 3 common signals)
		1/3	3	
			4	104 (26 segment signals, 4 common signals)
		1/4	8	176 (22 segment signals, 8 common signals)
	Internal voltage boosting	1/3	3	78 (26 segment signals, 3 common signals)
			4	104 (26 segment signals, 4 common signals)
		1/4	8	176 (22 segment signals, 8 common signals)
	Capacitor split	1/3	3	78 (26 segment signals, 3 common signals)
			4	104 (26 segment signals, 4 common signals)
Waveform B	External resistance division, internal voltage boosting	1/3	4	
		1/4	8	176 (22 segment signals, 8 common signals)
	Capacitor split	1/3	4	104 (26 segment signals, 4 common signals)

(b) 64-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	39 (39 segment signals, 1 common signal)
		1/2	2	78 (39 segment signals, 2 common signals)
			3	117 (39 segment signals, 3 common signals)
		1/3	3	
			4	156 (39 segment signals, 4 common signals)
		1/4	8	280 (35 segment signals, 8 common signals)
	Internal voltage boosting	1/3	3	117 (39 segment signals, 3 common signals)
			4	156 (39 segment signals, 4 common signals)
		1/4	8	280 (35 segment signals, 8 common signals)
	Capacitor split	1/3	3	117 (39 segment signals, 3 common signals)
			4	156 (39 segment signals, 4 common signals)
Waveform B	External resistance division, internal voltage boosting	1/3	4	
		1/4	8	280 (35 segment signals, 8 common signals)
	Capacitor split	1/3	4	156 (39 segment signals, 4 common signals)

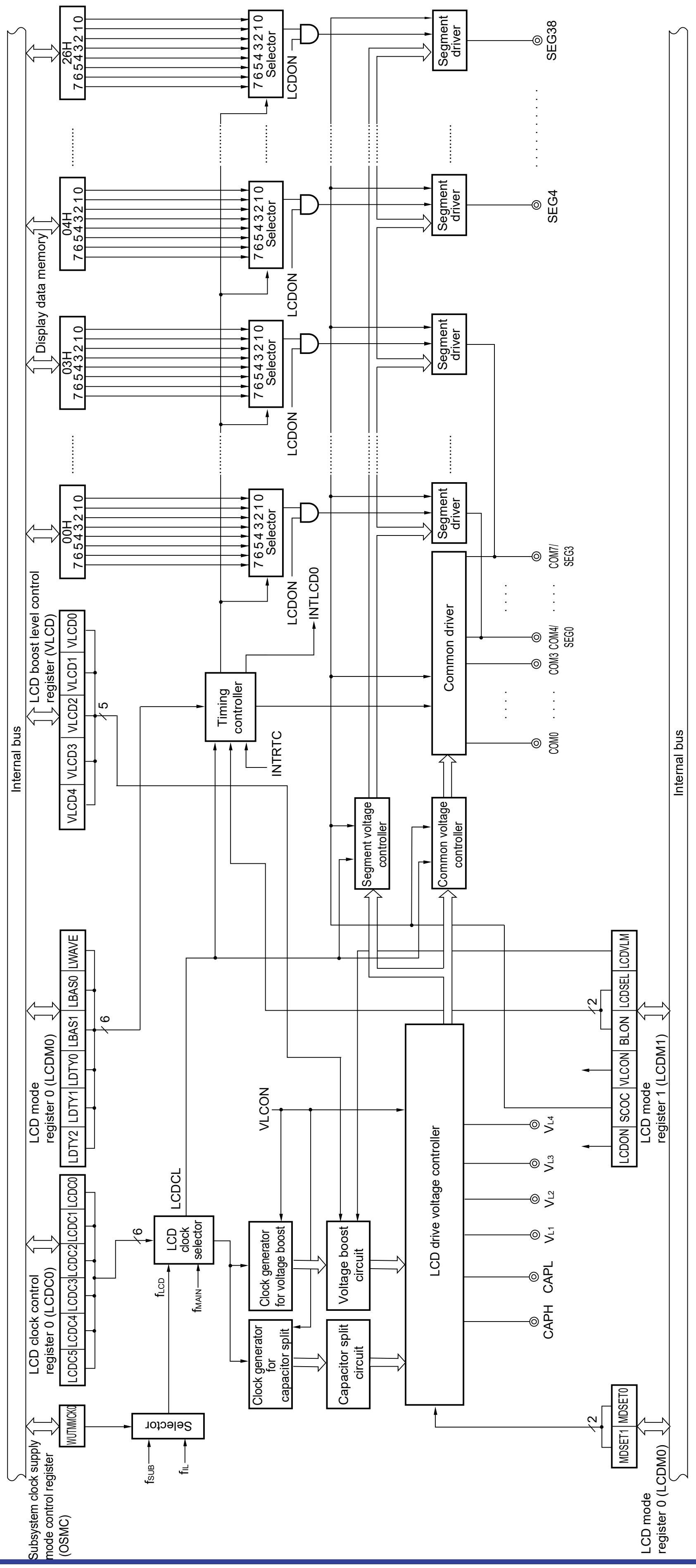
13.2 Configuration of LCD Controller/Driver

The LCD controller/driver consists of the following hardware.

Table 13-3. Configuration of LCD Controller/Driver

Item	Configuration
Control registers	Peripheral enable register 0 (PER0) 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 4 (PFSEG0 to PFSEG4) Port mode registers 1, 3 to 7, 12, 14 (PM1, PM3 to PM7, PM12, PM14)

Figure 13-1. Block Diagram of LCD Controller/Driver



13.3 Registers Controlling LCD Controller/Driver

The following ten registers are used to control the LCD controller/driver.

- Peripheral enable register 0 (PER0)
- 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 4 (PFSEG0 to PFSEG4)
- Port mode registers 1, 3 to 7, 12, 14 (PM1, PM3 to PM7, PM12, PM14)

13.3.1 Peripheral enable register 0 (PER0)

PER0 enables or disables 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 LCD controller/driver is used in subsystem clock (f_{SUB}), be sure to set bit 7 (RTCEN) of this register to 1.

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

Reset signal generation sets this register to 00H.

Figure 13-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

RTCEN	Real-time clock (RTC), 12-bit interval timer	LCD controller/driver, clock output/buzzer output controller	
		When subsystem clock (f_{SUB}) is selected	When subsystem clock (f_{SUB}) is not selected
0	Stops input clock supply. ● SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. ● The real-time clock (RTC) and 12-bit interval timer are in the reset status.	Stops input clock and subsystem clock supply. ● SFR used by the LCD controller/driver and clock output/buzzer output controller can be read and written.	Enables input clock and main system clock supply ● SFR used by the LCD controller/driver and clock output/buzzer output controller can be read and written.
1	Enables input clock supply. ● SFR used by the real-time clock (RTC) and 12-bit interval timer can be read and written.	Enables input clock and subsystem clock supply ● SFR used by the LCD controller/driver and clock output/buzzer output controller can be read and written.	

Cautions 1. The subsystem clock supply to peripheral functions other than the real-time clock, 12-bit interval timer, and LCD controller/driver can be stopped in STOP mode and HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the subsystem clock supply mode control register (OSMC) to 1. In this case, set the RTCEN bit of the PER0 register to 1 and the other bits (bits 0 to 6) to 0.

2. Be sure to set bits 1, 3, 4 and 6 to “0”.

13.3.2 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 13-3. Format of LCD Mode Register 0 (LCDM0) (1/2)

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

Figure 13-3. Format of LCD Mode Register 0 (LCDM0) (2/2)

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

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

Cautions 1. Do not rewrite the LCDM0 value while the SCOC bit of the LCDM1 register = 1.

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.

3. Only the combinations of display waveform, number of time slices, and bias method shown in Table 13-4 are supported.

Combinations of settings not shown in Table 13-4 are prohibited.

Table 13-4. Combinations of Display Waveform, Time Slices, and Bias Method and Frame Frequency

Display Mode			Set Value						Driving Voltage Generation Method		
Display Waveform	Number of Time Slices	Bias Mode	LWAVE	LDTY2	LDTY1	LDTY0	LBAS1	LBAS0	External Resistance Division	Internal Voltage Boosting	Capacitor Split
Waveform A	8	1/4	0	1	0	1	1	0	O (24 to 128 Hz)	O (24 to 64 Hz)	×
Waveform A	4	1/3	0	0	1	1	0	1	O (24 to 128 Hz)	O (24 to 128 Hz)	O (24 to 128 Hz)
Waveform A	3	1/3	0	0	1	0	0	1	O (32 to 128 Hz)	O (32 to 128 Hz)	O (32 to 128 Hz)
Waveform A	3	1/2	0	0	1	0	0	0	O (32 to 128 Hz)	×	×
Waveform A	2	1/2	0	0	0	1	0	0	O (24 to 128 Hz)	×	×
Waveform A	Static		0	0	0	0	0	0	O (24 to 128 Hz)	×	×
Waveform B	8	1/4	1	1	0	1	1	0	O (24 to 128 Hz)	O (24 to 64 Hz)	×
Waveform B	4	1/3	1	0	1	1	0	1	O (24 to 128 Hz)	O (24 to 128 Hz)	O (24 to 128 Hz)

Remark **O**: Supported

×: Not supported

13.3.3 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 13-4. Format of LCD Mode Register 1 (LCDM1) (1/2)

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 (RTC))
1	1	

Notes 1. Setting is prohibited when External resistance division method.

2. When f_{IL} is selected as the LCD source clock (f_{LCD}), be sure to set the BLON bit to “0”.

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

Figure 13-4. Format of LCD Mode Register 1 (LCDM1) (2/2)

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

LCDVLM[Note]	Control of default value of voltage boosting pin
0	Set when $V_{DD} \geq 2.7$ V
1	Set when $V_{DD} \leq 4.2$ V

Note This bit is used to boost the voltage efficiently when using the voltage boost circuit by setting the initial V_{LX} pin status.
If V_{DD} is 2.7 V or higher when voltage boosting starts, set the LCDVLM bit to 0; if V_{DD} is 4.2 V or lower, set the LCDVLM bit to 1.
If V_{DD} is within the range between 2.7 V and 4.2 V, the LCDVLM bit may be set to 0 or 1.

Cautions 1. To reduce power consumption when nothing is to be displayed on the LCD while the voltage boost circuit is in use, set the SCOC and VLCON bits to “0”, and set the MDSET1 and MDSET0 bits to “00”.
When MDSET1 and MDSET0 = 01, the internal reference voltage generator operates and so consumes power.

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.

3. Do not rewrite the VLCON and LCDVLM bits while SCOC = 1.

4. Set the BLON and LCDSEL bits to 0 when 8 has been selected as the number of time slices for the display mode.

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.

13.3.4 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, 12-bit interval timer, clock output/buzzer output, and LCD controller/driver, is stopped in STOP mode and HALT mode while the subsystem clock is selected as the CPU clock. Set bit 7 (RTCEN) of peripheral enable register 0 (PER0) to 1 before specifying this setting.

In addition, the OSMC register can be used to select the operation clock of the real-time clock, 12-bit interval timer, 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 13-5. 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 HALT mode while subsystem clock is selected as CPU clock
0	Enables subsystem clock supply to peripheral functions. (See Table 19-1 for the peripheral functions whose operations are enabled.)
1	Stops subsystem clock supply to peripheral functions except real-time clock, 12-bit interval timer, and LCD controller/driver.

WUTMMCK0[Note]	Selection of operation clock for real-time clock, 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, 12-bit interval timer, and LCD driver/controller are all stopped. The setting must not be changed after the operation starts.

These are stopped as follows:

Real-time clock: 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)

13.3.5 LCD clock control register 0 (LCDC0)

LCDC0 specifies the LCD source clock and 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 13-6. Format of LCD Clock Control Register 0 (LCDC0) (1/2)

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

(1/2)

LCDC05	LCDC04	LCDC03	LCDC02	LCDC01	LCDC00	LCD clock (LCDCL)
0	0	0	0	0	1	$f_{SUB}/2^2$ or $f_{IL}/2^2$
0	0	0	0	1	0	$f_{SUB}/2^3$ or $f_{IL}/2^3$
0	0	0	0	1	1	$f_{SUB}/2^4$ or $f_{IL}/2^4$
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	$f_{SUB}/2^{10}$

Cautions 1. Be sure to set bits 6 and 7 to “0”.

2. Set the frame frequency in a range from 32 Hz to 128 Hz (24 Hz to 128 Hz when f_{IL} is selected). And set the LCD clock (LCDCL) to no more than 512 Hz (no more than 235 Hz when f_{IL} is selected) when the internal voltage boosting method, and the capacitor split method have been specified.

3. Do not set LCDC0 when the SCOC bit of the LCDM1 register is 1.

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

Figure 13-6. Format of LCD Clock Control Register 0 (LCDC0) (2/2)

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

(2/2)

LCDC05	LCDC04	LCDC03	LCDC02	LCDC01	LCDC00	LCD clock (LCDCL)
0	1	0	0	0	1	$f_{MAIN}/2^8$
0	1	0	0	1	0	$f_{MAIN}/2^9$
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

Cautions 1. Do not set LCDC0 when the SCOC bit of the LCDM1 register is 1.

2. Be sure to set bits 6 and 7 to “0”.

3. When the internal voltage boosting method and the capacitor split method are specified, set the LCD clock (LCDCL) as follows. For details, see Table 13-4 Combinations of Display Waveform, Time Slices, and Bias Method and Frame Frequency

- **If f_{SUB} is selected, set the clock to a frequency no greater than 512 Hz.**
- **If f_{IL} is selected, set the clock to a frequency no greater than 235 Hz.**

Remark f_{MAIN}: Main system clock frequency

f_{SUB}: Subsystem clock frequency

f_{IL}: Low-speed on-chip oscillator clock frequency

13.3.6 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 13-7. 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		

Cautions 1. The VLCD setting is valid only when the voltage boost circuit is operating.

2. Be sure to set bits 5 to 7 to “0”.

3. Be sure to change the VLCD value after having stopped the operation of the voltage boost circuit (VLCON = 0).

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.

5. With the external resistance division method and the capacitor split method, use the default value (04H) for the VLCD resistor.

13.3.7 LCD input switch control register (ISCLCD)

Input to the schmitt trigger buffer must be invalid 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 13-8. 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

Cautions 1. If ISCVL3 = 0, set the corresponding port registers as follows:
PU125 bit of PU12 register = 0, P125 bit of P12 register = 0
2. If 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

(a) 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).

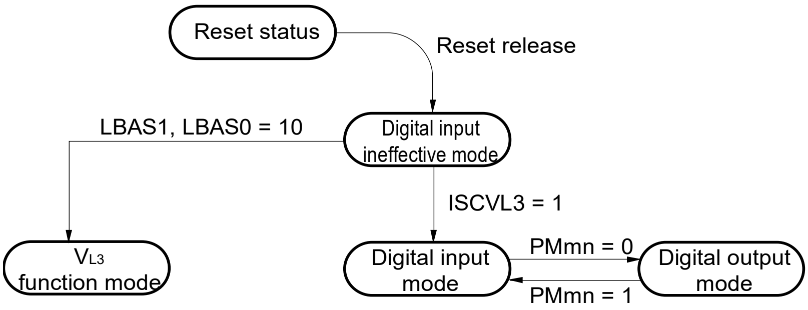
- **V_{L3}/P125**

Table 13-5. 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 13-9. 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).

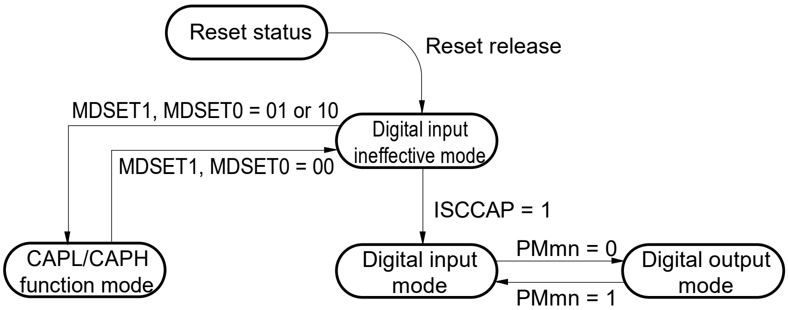
- **CAPL/P126 and CAPH/P127**

Table 13-6. 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 13-10. 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).

13.3.8 LCD port function registers 0 to 4 (PFSEG0 to PFSEG4)

These registers specify whether to use pins P10 to P17, P30 to P32, P41 to P43, P50 to P54, P60, P61, P70 to P74, P120, and P140 to P147 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.

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 13-7 Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits).

Figure 13-11. Format of LCD Port Function Registers 0 to 4 (64-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: 7FH R/W

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

PFSEGxx (xx = 04 to 46)	Port (other than segment output)/segment outputs specification of Pmn pins (mn = 10 to 17, 30 to 32, 41 to 43, 50 to 54, 60, 61, 70 to 74, 120, 140 to 147)
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 13-7. Segment Output Pins in Each Product and Correspondence with PFSEG Register (PFSEG Bits)

Bit Name of PFSEG Register	Corresponding SEGxx Pins	Alternate Port	64-pin	48-pin
PFSEG04	SEG4	P15	√	√
PFSEG05	SEG5	P16	√	√
PFSEG06	SEG6	P17	√	√
PFSEG07	SEG7	P50	√	√
PFSEG08	SEG8	P51	√	–
PFSEG09	SEG9	P52	√	–
PFSEG10	SEG10	P53	√	–
PFSEG11	SEG11	P54	√	–
PFSEG12	SEG12	P74	√	–
PFSEG13	SEG13	P73	√	–
PFSEG14	SEG14	P72	√	–
PFSEG15	SEG15	P71	√	–
PFSEG16	SEG16	P70	√	√
PFSEG17	SEG17	P32	√	√
PFSEG18	SEG18	P31	√	√
PFSEG19	SEG19	P30	√	√
PFSEG20	SEG20	P61	√	√
PFSEG21	SEG21	P60	√	√
PFSEG22	SEG22	P43	√	–
PFSEG23	SEG23	P42	√	–
PFSEG24	SEG24	P41	√	√
PFSEG25	SEG25	P120	√	√
PFSEG26	SEG26	P141	√	√
PFSEG27	SEG27	P140	√	√
PFSEG28	SEG28	P10	√	√
PFSEG29	SEG29	P11	√	√
PFSEG30	SEG30	P12	√	√
PFSEG31	SEG31	P13	√	√
PFSEG32	SEG32	P14	√	√
PFSEG33	SEG33	P142	√	√
PFSEG34	SEG34	P143	√	√
PFSEG35	SEG35	P144	√	√
PFSEG36	SEG36	P145	√	–
PFSEG37	SEG37	P146	√	–
PFSEG38	SEG38	P147	√	–

(a) 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 4 (PFSEG0 to PFSEG4).

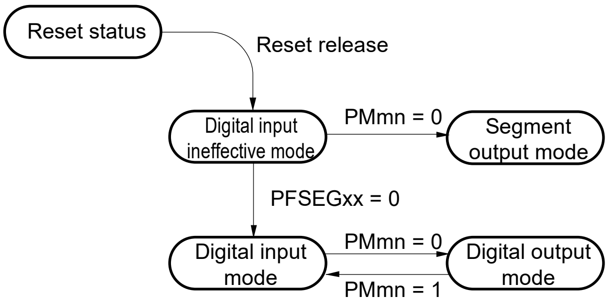
- **P10 to P12, P15 to P17, P30 to P32, P42, P43, P50 to P54, P60, P61, P70 to P74, P140, P141 (ports that do not serve as analog input pins (ANIxx))**

Table 13-8. Settings of SEGxx/Port Pin Function

PFSEGxx Bit of PFSEG0 to PFSEG4 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 13-12. 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).

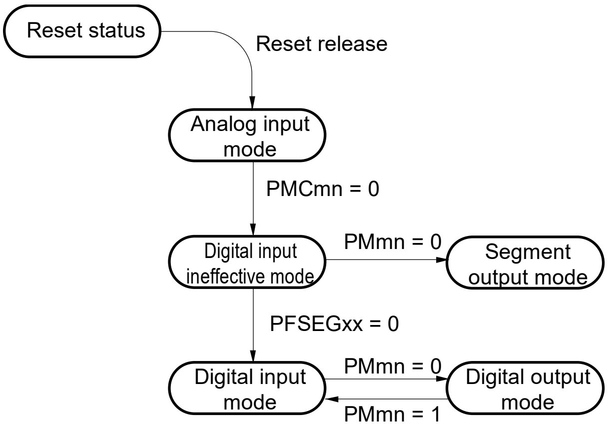
- **P13, P14, P41, P120, P142 to P147 (ports that serve as analog input pins (ANIxx))**

Table 13-9. Settings of ANIxx/SEGxx/Port Pin Function

PMCxx Bit of PMCxx Register	PFSEGxx Bit PFSEG0 to PFSEG4 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 13-13. 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).

13.3.9 Port mode registers 1, 3 to 7, 12, 14 (PM1, PM3 to PM7, PM12, PM14)

These registers specify input/output of ports 1, 3 to 7, 12, and 14 in 1-bit units.

When using the ports (such as P10/SCK00/SEG28, P120/ANI17/SEG25) 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 P10/SCK00/SEG28 for segment output
Set the PM10 bit of port mode register 1 to “0”.
Set the P10 bit of port register 1 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 13-14. Format of Port Mode Registers 1, 3 to 7, 12, 14 (PM1, PM3 to PM7, PM12, PM14) (64-pin Products)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM3	1	1	1	1	1	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	1	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26H	FFH	R/W
PM7	1	1	1	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM12	PM127	PM126	PM125	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	PM147	PM146	PM145	PM144	PM143	PM142	PM141	PM140	FFF2EH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 1, 3 to 7, 12, 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 1, 3 to 7, 12, and 14 of the 64-pin products. The format of the port mode register of other products, see **Table 4-3 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product**.

13.4 LCD Display Data Registers

The LCD display data registers are mapped as shown in Table 13-10. The contents displayed on the LCD can be changed by changing the contents of the LCD display data registers.

Table 13-10. Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (1/2)

(a) Other than 8-time-slice (static, 2-time slice, 3-time slice, and 4-time slice)

Register Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	64-pin	48-pin
		COM 7	COM 6	COM 5	COM 4	COM 3	COM 2	COM 1	COM 0		
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)				√	√
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)				√	–

Remark √: Supported, –: Not supported

Table 13-10. Relationship Between LCD Display Data Register Contents and Segment/Common Outputs (2/2)

(b) 8-time slice

Register Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	64-pin	48-pin
		COM 7	COM 6	COM 5	COM 4	COM 3	COM 2	COM 1	COM 0		
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								√	–

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.

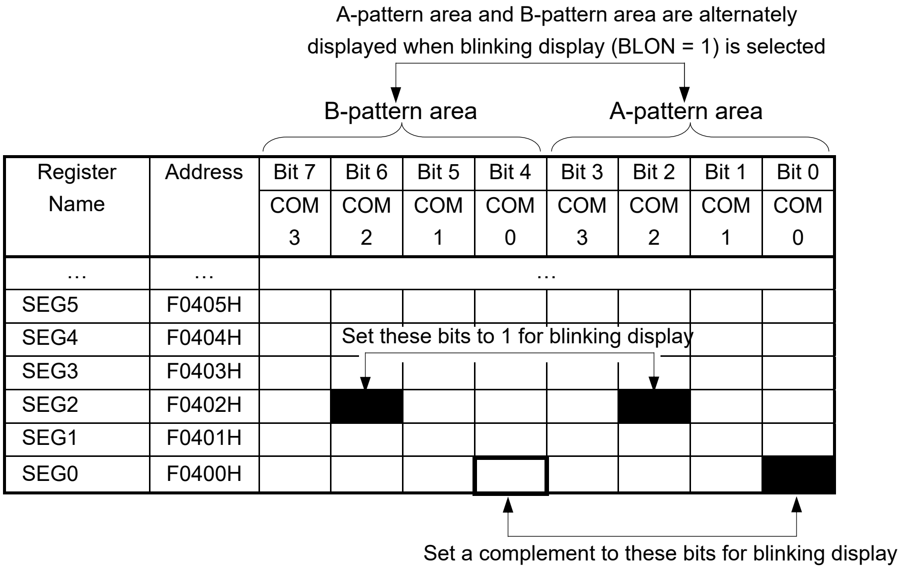
13.5 Selection of LCD Display Register

With R7F0C001G/L, R7F0C002G/L, 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 (RTC))

Caution When the number of time slices is eight, LCD display data registers (A-pattern, B-pattern, or blinking display) cannot be selected.

Figure 13-15. Example of Setting LCD Display Registers When Pattern Is Changed



A-pattern area and B-pattern area are alternately displayed when blinking display (BLON = 1) is selected

B-pattern area

A-pattern area

Register Name	Address	Bit 7 COM 3	Bit 6 COM 2	Bit 5 COM 1	Bit 4 COM 0	Bit 3 COM 3	Bit 2 COM 2	Bit 1 COM 1	Bit 0 COM 0
…	…	…							
SEG5	F0405H								
SEG4	F0404H								
SEG3	F0403H								
SEG2	F0402H		■				■		
SEG1	F0401H								
SEG0	F0400H				□				■

Set these bits to 1 for blinking display

Set a complement to these bits for blinking display

13.5.1 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 **13.4 LCD Display Data Registers** about the display area.

13.5.2 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 (RTC). See **CHAPTER 7 REAL-TIME CLOCK** 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 00H, 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 **13.4 LCD Display Data Registers** about the display area.

Next, the timing operation of display switching is shown.

Figure 13-16. Switching Operation from A-Pattern Display to Blinking Display

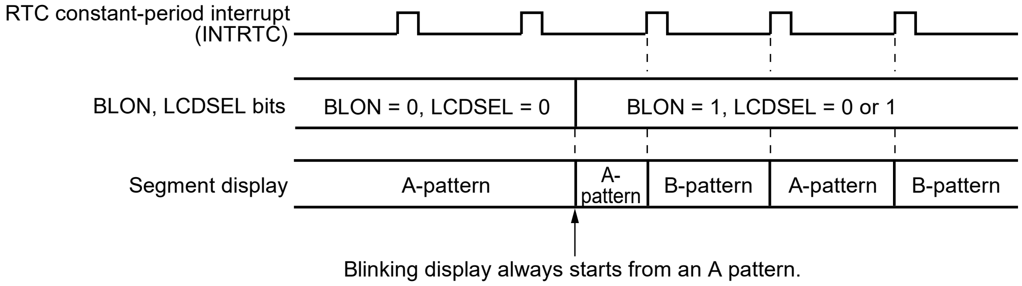
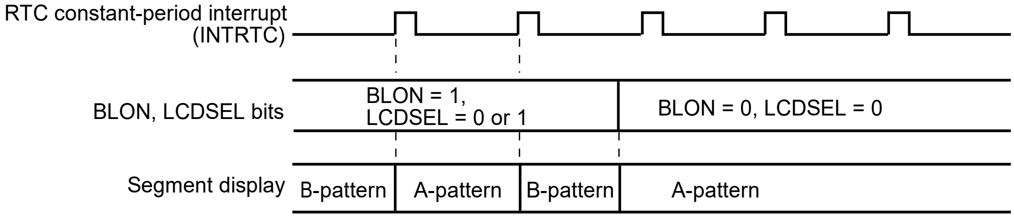


Figure 13-17. Switching Operation from Blinking Display to A-Pattern Display



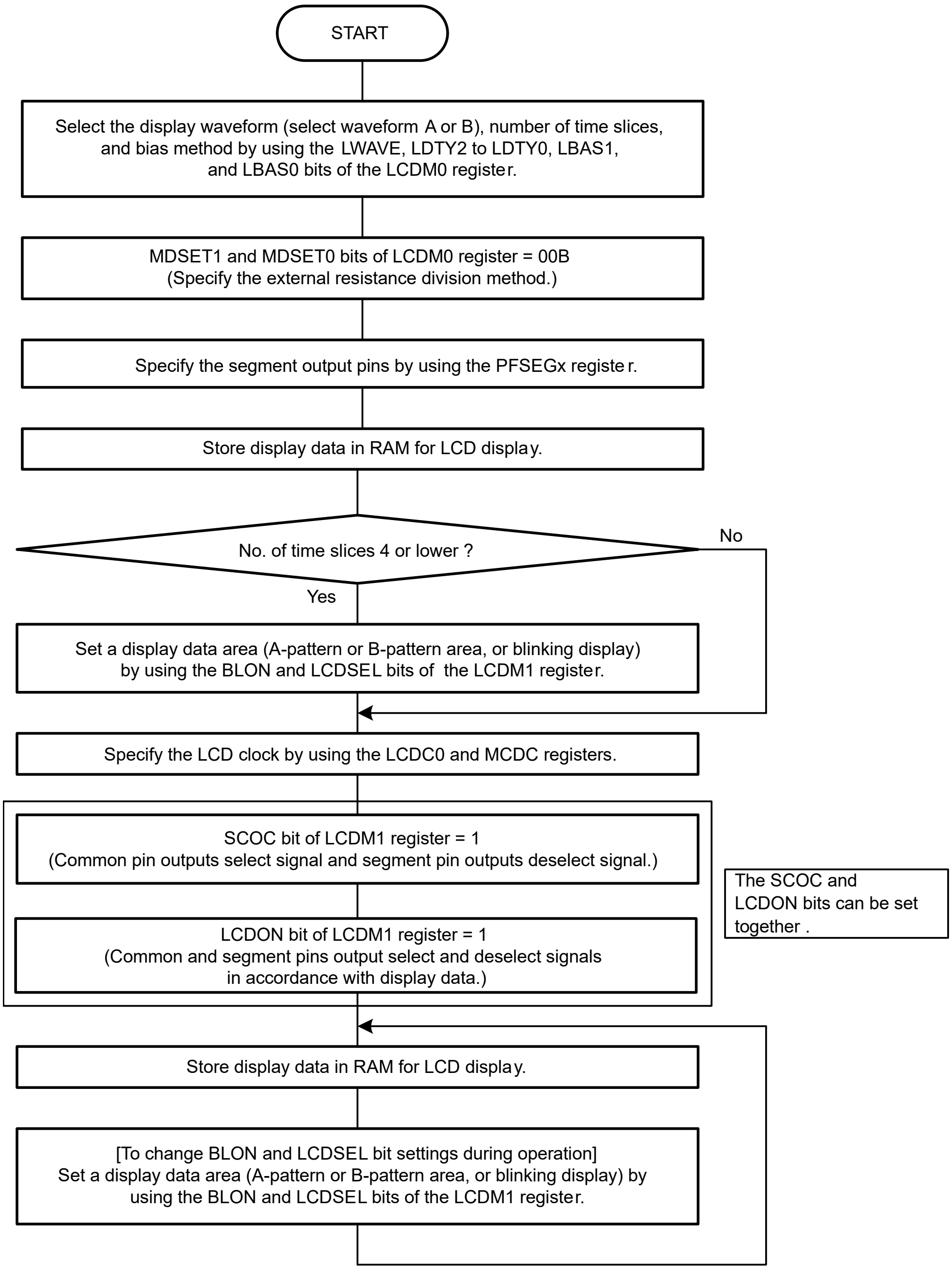
13.6 Setting the LCD Controller/Driver

Set the LCD controller/driver using the following procedure.

Caution To operate the LCD controller/driver, be sure to follow procedures (1) to (3). Unless these procedures are observed, the operation will not be guaranteed.

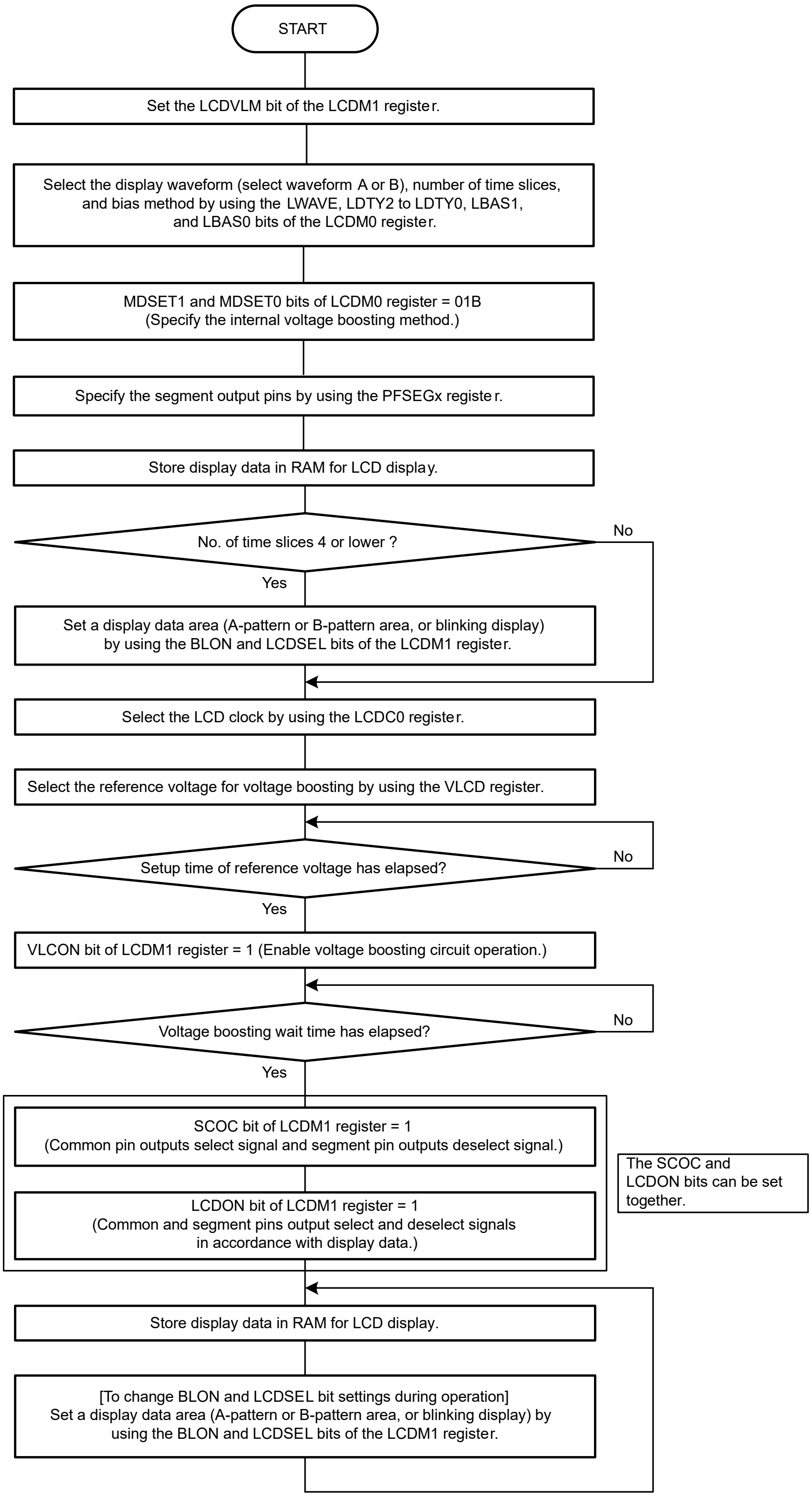
(1) External resistance division method

Figure 13-18. External Resistance Division Method Setting Procedure



(2) Internal voltage boosting method

Figure 13-19. Internal Voltage Boosting Method Setting Procedure

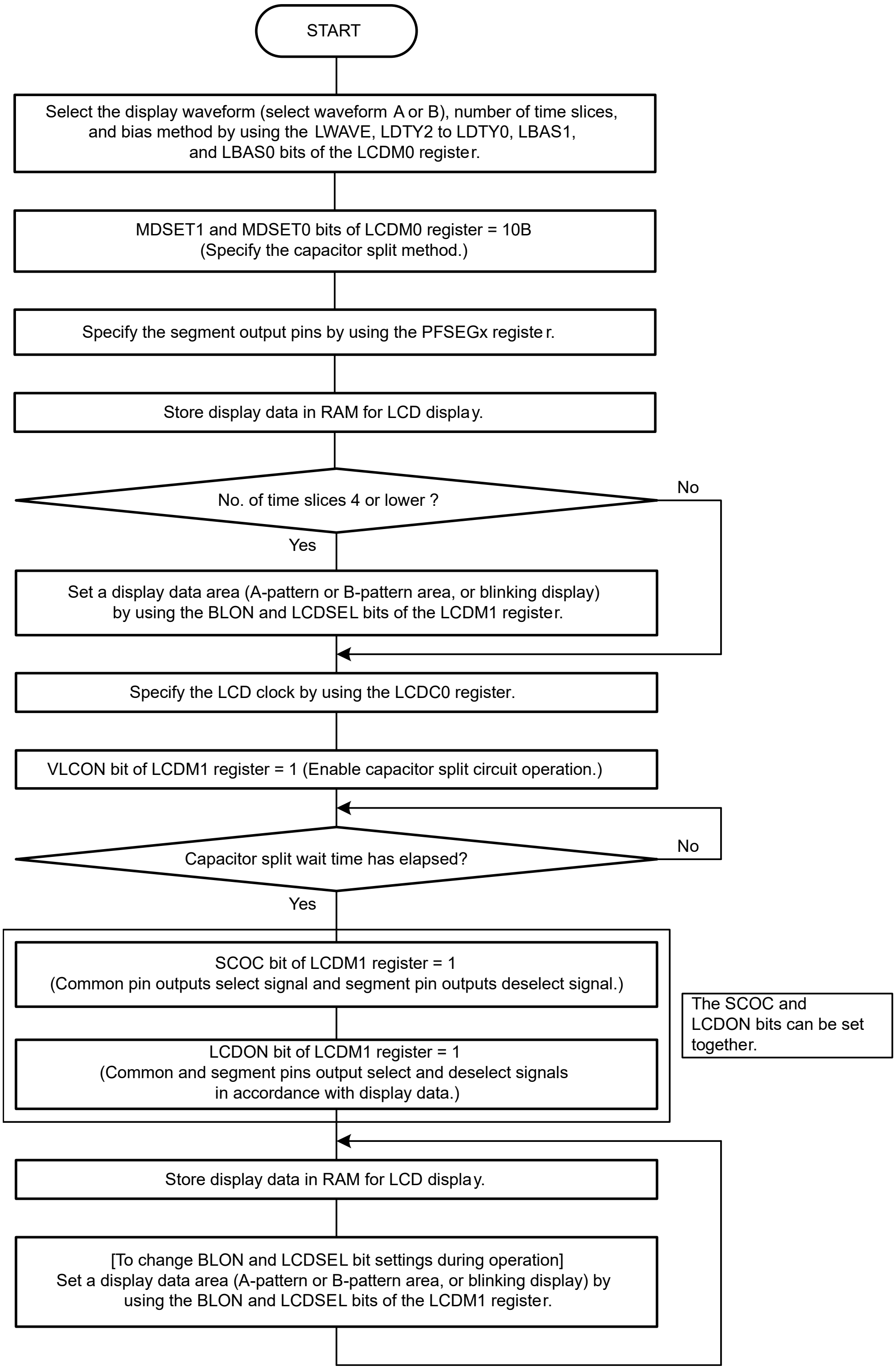


Cautions 1. Wait until the setup time has elapsed even if not changing the setting of the VLCD register.

2. For the specifications of the reference voltage setup time and voltage boosting wait time, see CHAPTER 27 ELECTRICAL SPECIFICATIONS.

(3) Capacitor split method

Figure 13-20. Capacitor Split Method Setting Procedure

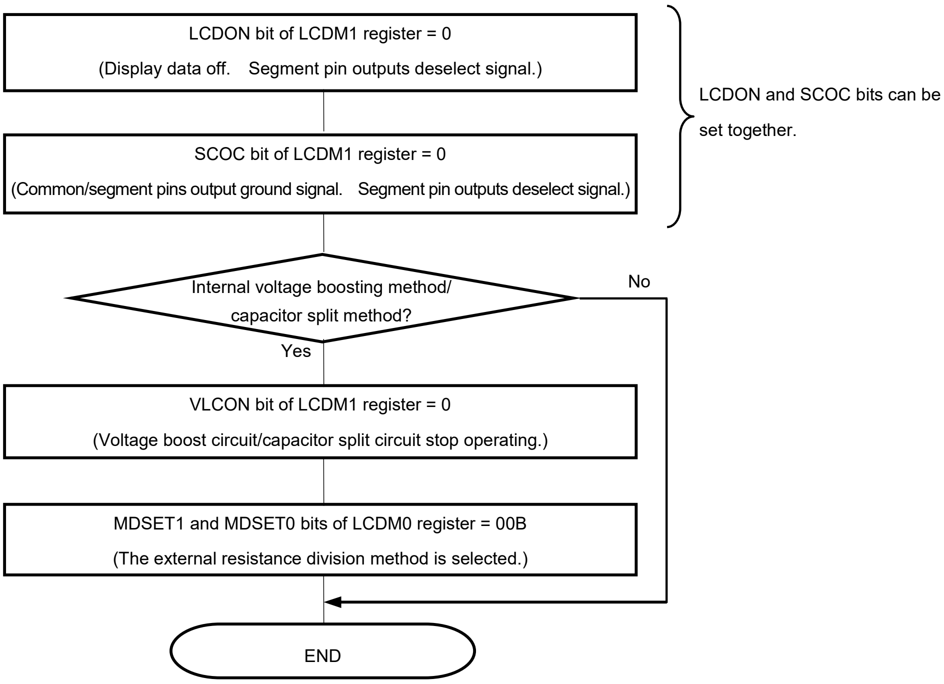


Caution For the specifications of the capacitor split wait time, see CHAPTER 27 ELECTRICAL SPECIFICATIONS.

13.7 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 13-21. Operation Stop Procedure



Caution **Stopping the voltage boost/capacitor split circuits is prohibited while the display is on (SCOC and LCDON bits of LCDM1 register = 00B). Otherwise, the operation will not be guaranteed. Be sure to turn off display (SCOC and LCDON bits of LCDM1 register = 00B) before stopping the voltage boost/capacitor split circuits (VLCON bit of LCDM1 register = 0).**

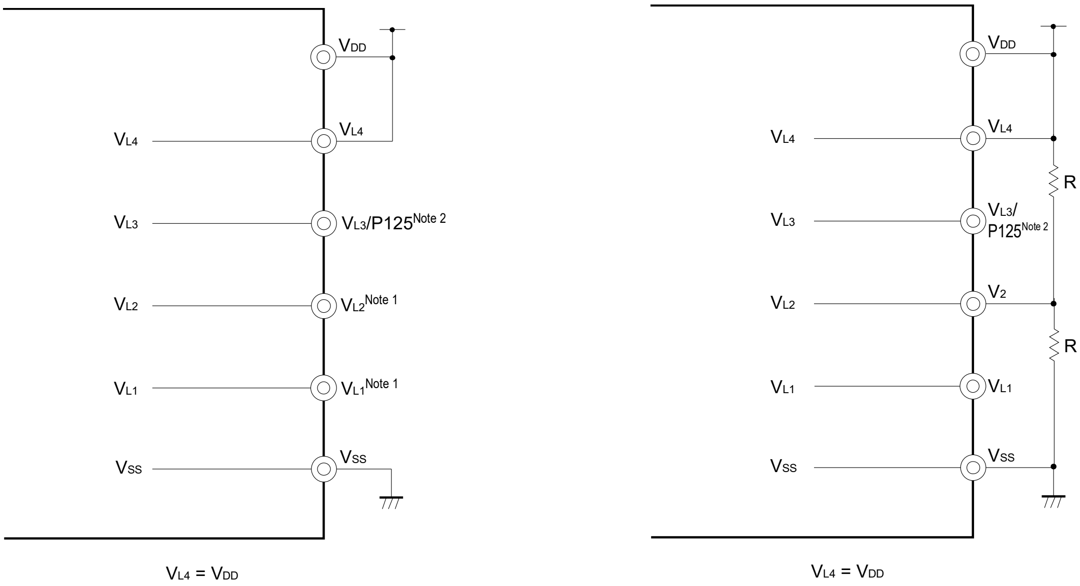
13.8 Supplying LCD Drive Voltages V_{L1}, V_{L2}, V_{L3}, and V_{L4}

The power supply voltages for the LCD driver can be produced through external resistance division, internal voltage boosting, or capacitor split.

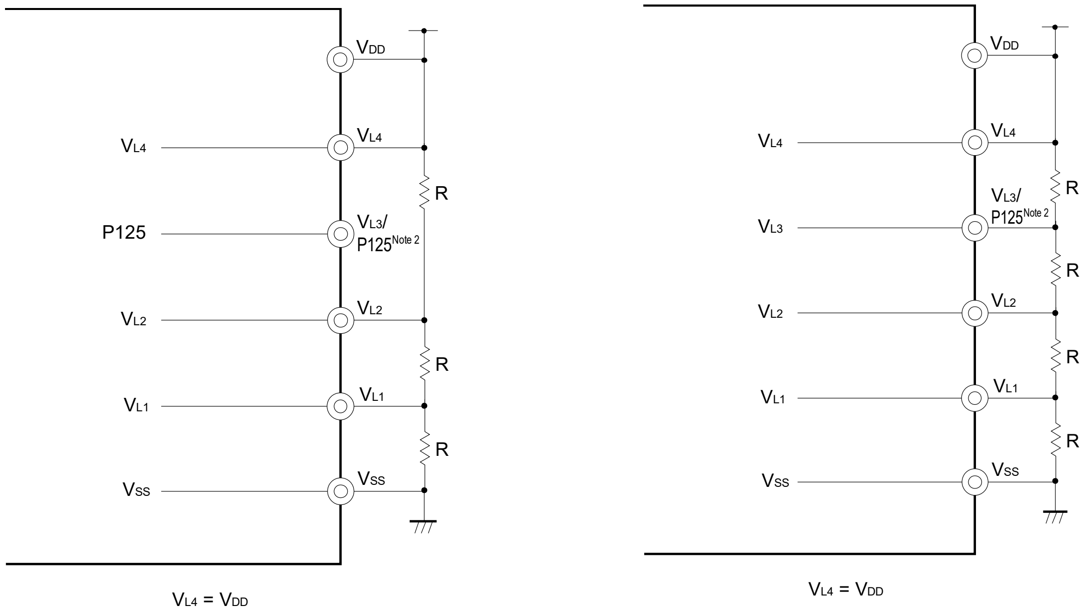
13.8.1 External resistance division method

Figure 13-22 shows examples of LCD drive voltage connection, corresponding to each bias method.

Figure 13-22. Examples of LCD Drive Power Connections (External Resistance Division Method)



(a) Static display mode **(b) 1/2 bias method**



(c) 1/3 bias method **(d) 1/4 bias method**

Notes 1. Connect V_{L1} and V_{L2} to GND or leave open.

2. 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.**

13.8.2 Internal voltage boosting method

R7F0C001G/L, R7F0C002G/L 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.

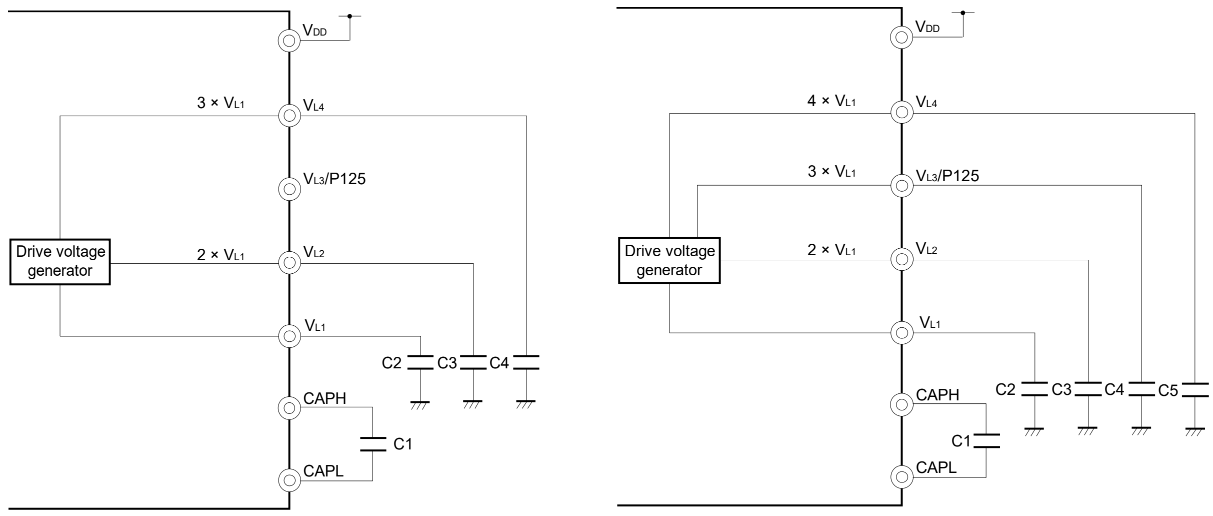
The LCD drive voltage of the internal voltage boosting method can supply a constant voltage, regardless of changes in V_{DD}, because it is a power supply separate from the main unit.

In addition, a contrast can be adjusted by using the LCD boost level control register (VLCD).

Table 13-11. 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 13-23. Examples of LCD Drive Power Connections (Internal Voltage Boosting Method)



(a) 1/3 bias method

(b) 1/4 bias method

Remark Use a capacitor with as little leakage as possible.
In addition, make C1 a nonpolar capacitor.

13.8.3 Capacitor split method

R7F0C001G/L, R7F0C002G/L 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.

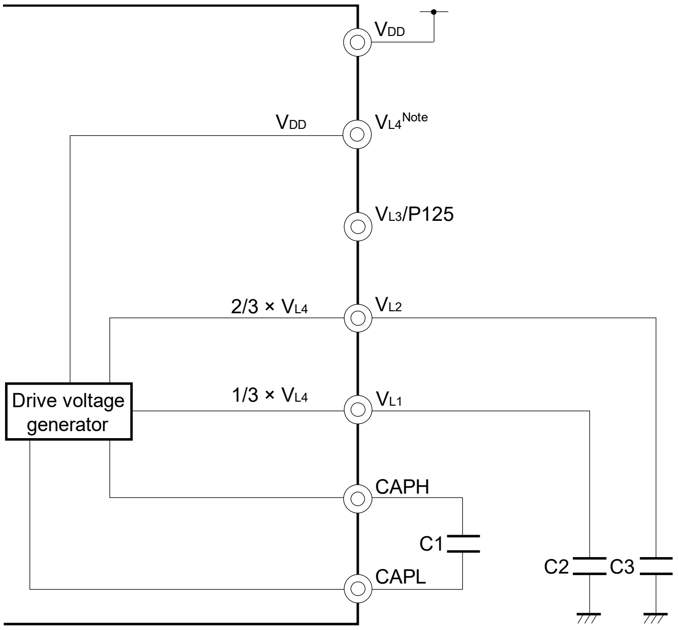
Different from the external resistance division method, there is always no current flowing with the capacitor split method, so current consumption can be reduced.

Table 13-12. LCD Drive Voltages (Capacitor Split Method)

Bias Method / LCD Drive Voltage Pin	1/3 Bias Method
V_{L4}	V_{DD}
V_{L3}	–
V_{L2}	2/3 × V_{L4}
V_{L1}	1/3 × V_{L4}

Figure 13-24. Examples of LCD Drive Power Connections (Capacitor Split Method)

- **1/3 bias method**



Note When switching to internal voltage boosting, connect the capacitor C4 as shown in **Figure 13-23 Examples of LCD Drive Power Connections (Internal Voltage Boosting Method)**.

Remark Use a capacitor with as little leakage as possible.
In addition, make C1 a nonpolar capacitor.

13.9 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 13-13. 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 13-13. 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 **13.4 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 SEG38).

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 SEG38).

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.

- 48-pin products: SEG0 to SEG7, SEG16 to SEG21, SEG24 to SEG35
- 64-pin products: SEG0 to SEG38

(3) Output waveforms of common and segment signals

The voltages listed in Table 13-14 are output as common and segment signals.

When both common and segment signals are at the select voltage, a display on-voltage of $\pm V_{LCD}$ is obtained. The other combinations of the signals correspond to the display off-voltage.

Table 13-14. LCD Drive Voltage

(a) Static display mode

Segment Signal Common 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

Segment Signal Common 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)

Segment Signal Common 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}	$-\frac{1}{3}V_{LCD}/+\frac{1}{3}V_{LCD}$	$+\frac{1}{3}V_{LCD}/-\frac{1}{3}V_{LCD}$

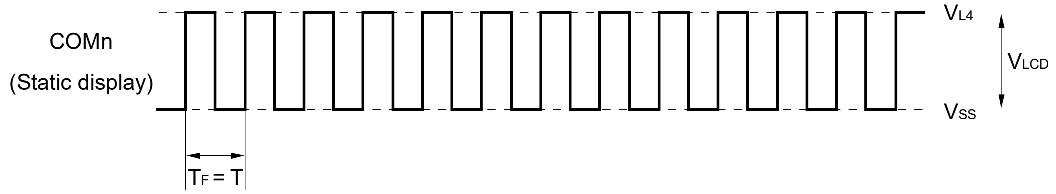
(d) 1/4 bias method (waveform A or B)

Segment Signal Common 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}	$-\frac{1}{4}V_{LCD}/+\frac{1}{4}V_{LCD}$	$+\frac{1}{4}V_{LCD}/-\frac{1}{4}V_{LCD}$

Figure 13-25 shows the common signal waveforms, and Figure 13-26 shows the voltages and phases of the common and segment signals.

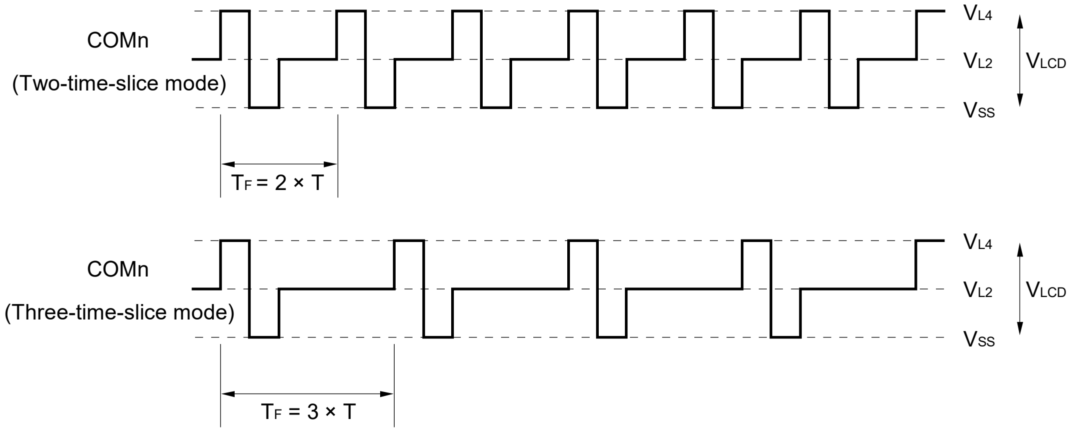
Figure 13-25. Common Signal Waveforms (1/2)

(a) Static display mode



T: One LCD clock period TF: Frame frequency

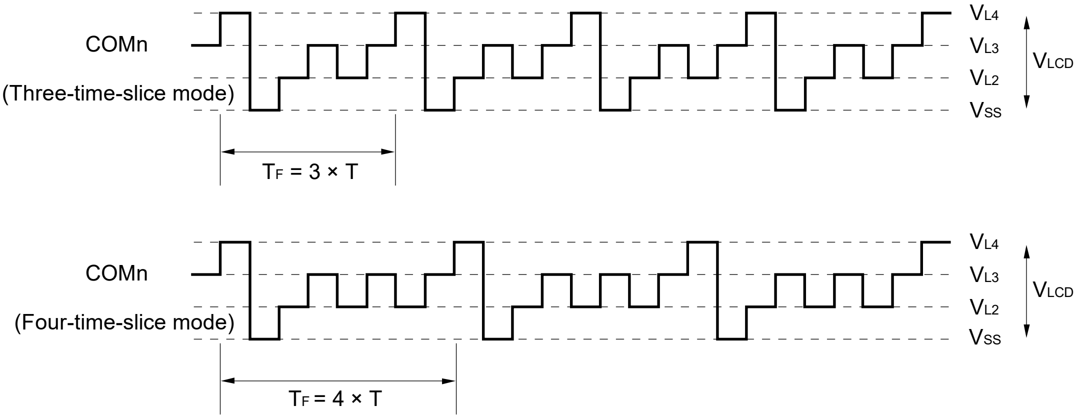
(b) 1/2 bias method



T: One LCD clock period TF: Frame frequency

Figure 13-25. Common Signal Waveforms (2/2)

(c) 1/3 bias method



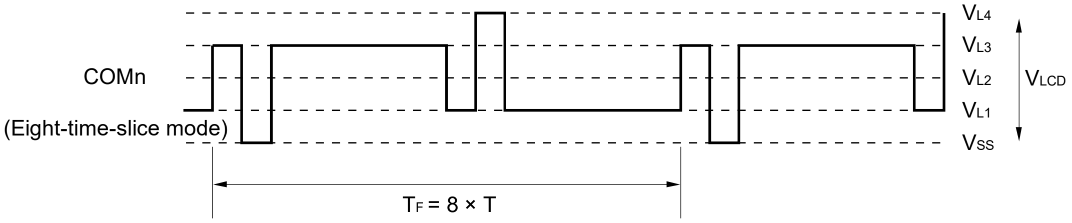
T: One LCD clock period T_F: Frame frequency

< 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 T_F: Frame frequency

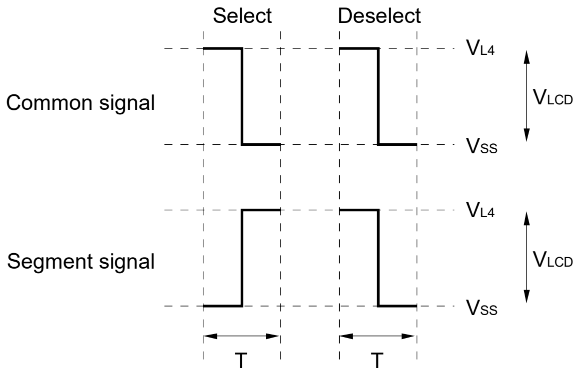
< Example of calculation of LCD frame frequency (When eight-time slot mode is used) >

LCD clock: $32768/2^8 = 256$ Hz (When setting to LCDC0 = 07H)

LCD frame frequency: 32 Hz

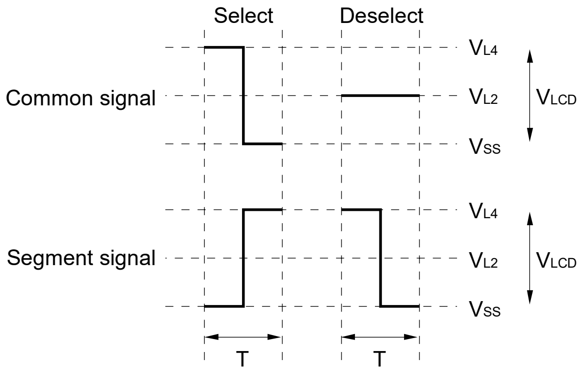
Figure 13-26. Voltages and Phases of Common and Segment Signals (1/3)

(a) Static display mode (waveform A)



T: One LCD clock period

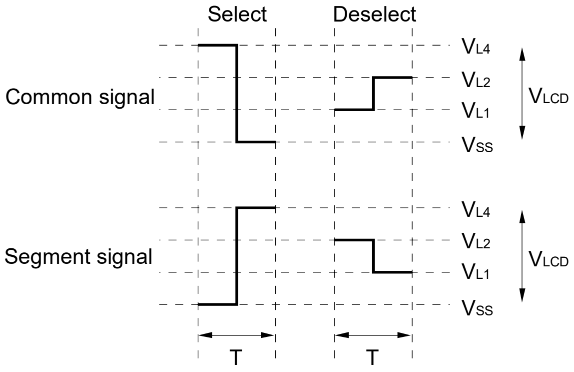
(b) 1/2 bias method (waveform A)



T: One LCD clock period

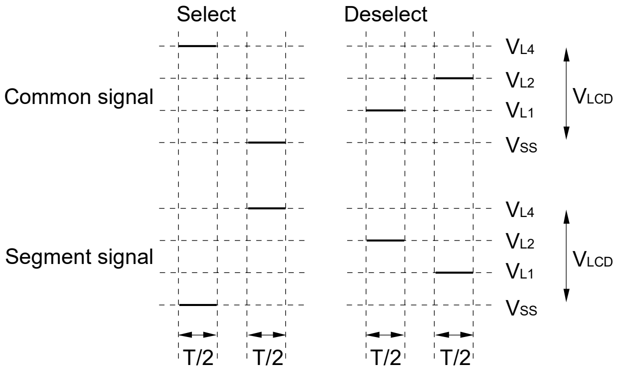
Figure 13-26. Voltages and Phases of Common and Segment Signals (2/3)

(c) 1/3 bias method (waveform A)



T: One LCD clock period

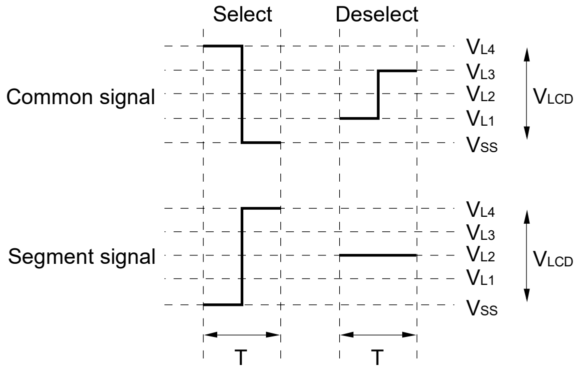
(d) 1/3 bias method (waveform B)



T: One LCD clock period

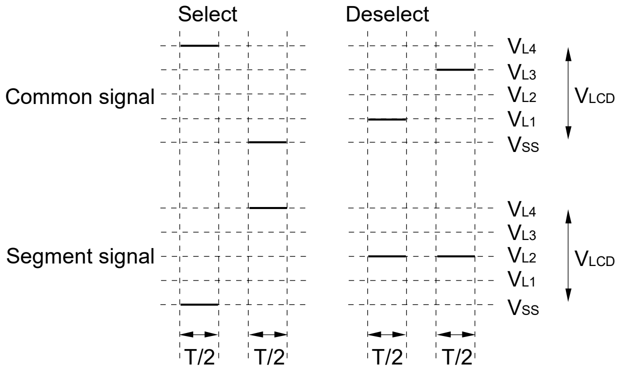
Figure 13-26. Voltages and Phases of Common and Segment Signals (3/3)

(e) 1/4 bias method (waveform A)



T: One LCD clock period

(f) 1/4 bias method (waveform B)



T: One LCD clock period

13.10 Display Modes

13.10.1 Static display example

Figure 13-28 shows how the three-digit LCD panel having the display pattern shown in Figure 13-27 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 13-15 at the timing of the common signal COM0; see Figure 13-27 for the relationship between the segment signals and LCD segments.

Table 13-15. Select and Deselect Voltages (COM0)

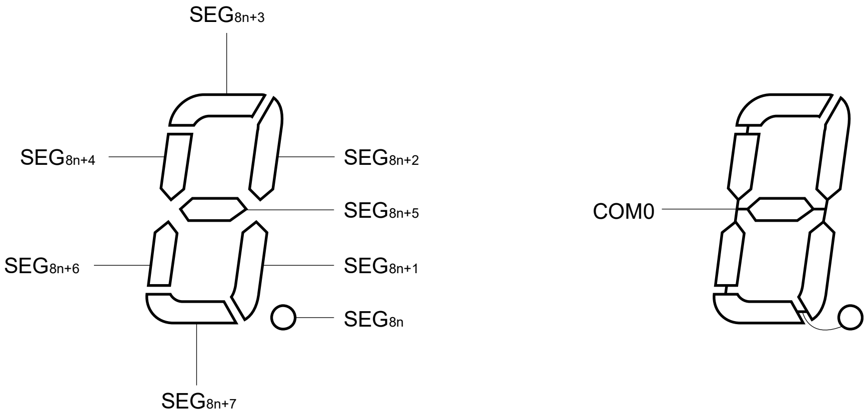
Segment / Common	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
COM0	Select	Deselect	Select	Select	Deselect	Select	Select	Select

According to Table 13-15, it is determined that the bit-0 pattern of the display data register locations (F0408H to F040FH) must be 10110111.

Figure 13-29 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.

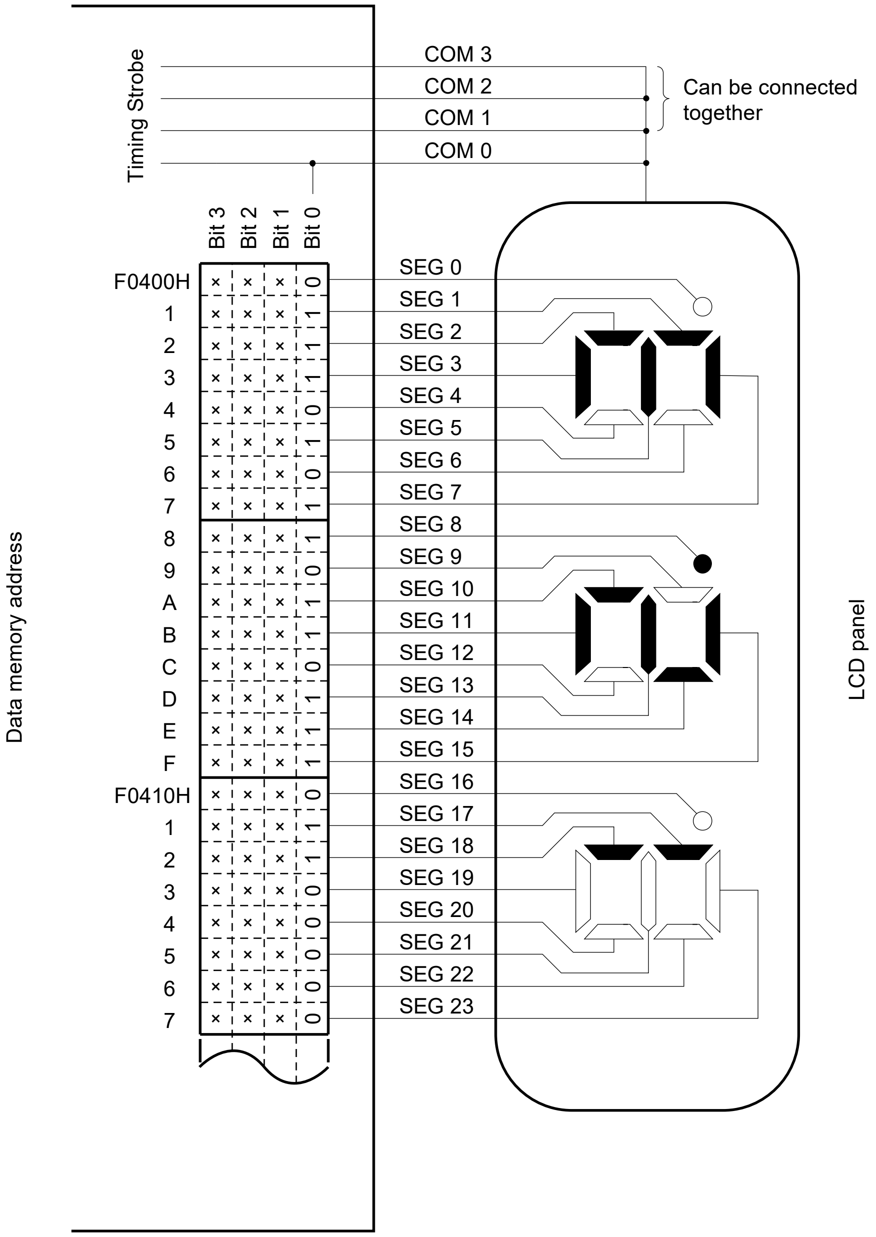
COM1 to COM3 are supplied with the same waveform as for COM0. So, COM0 to COM3 may be connected together to increase the driving capacity.

Figure 13-27. Static LCD Display Pattern and Electrode Connections



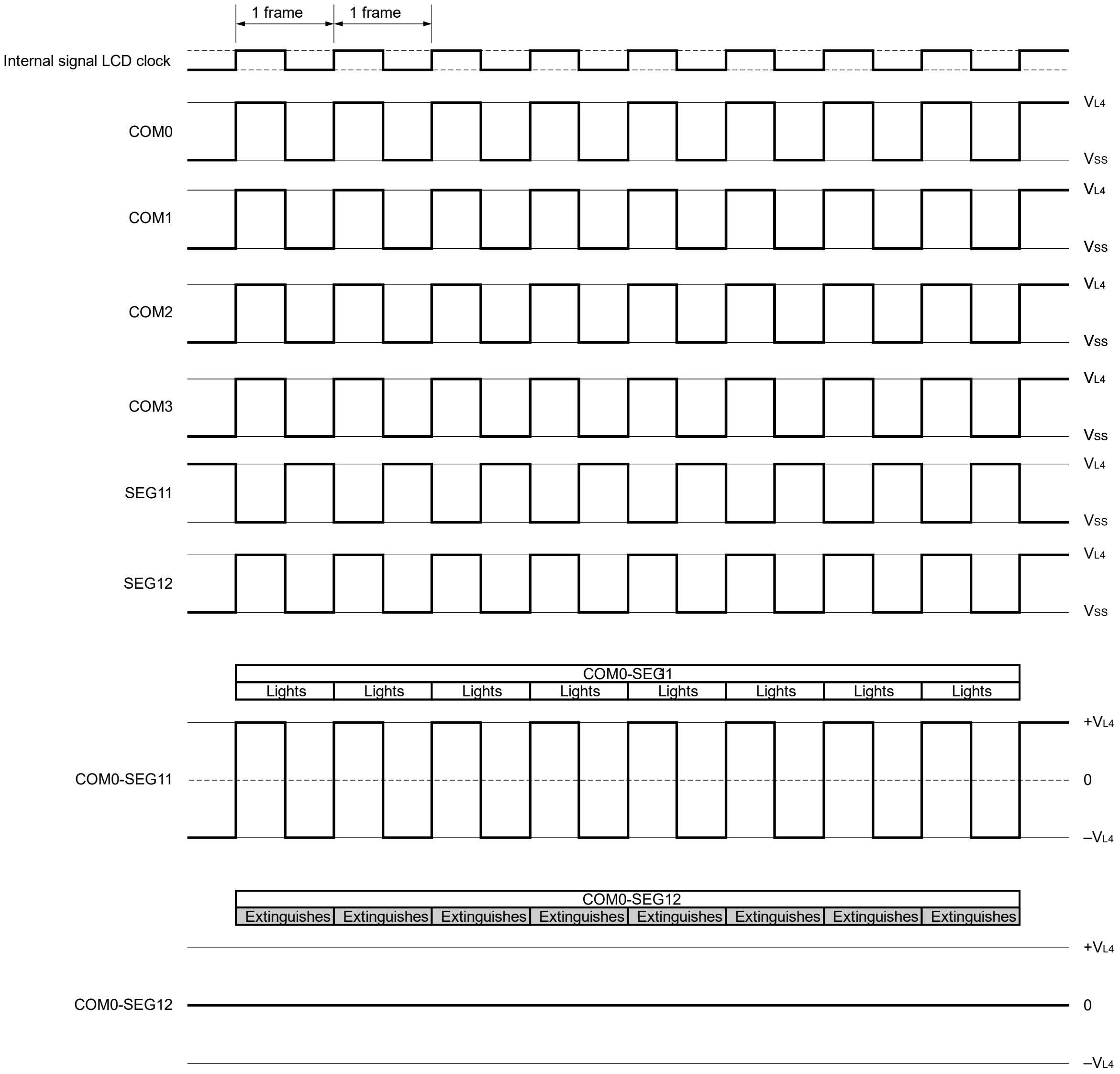
Remark 48-pin products: n = 0 to 2
64-pin products: n = 0 to 3

Figure 13-28. Example of Connecting Static LCD Panel



Remark ×: Don't care.

Figure 13-29. Static LCD Drive Waveform Examples for SEG11, SEG12, and COM0



13.10.2 Two-time-slice display example

Figure 13-31 shows how the 6-digit LCD panel having the display pattern shown in Figure 13-30 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” (∃) 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 13-16 at the timing of the common signals COM0 and COM1; see Figure 13-30 for the relationship between the segment signals and LCD segments.

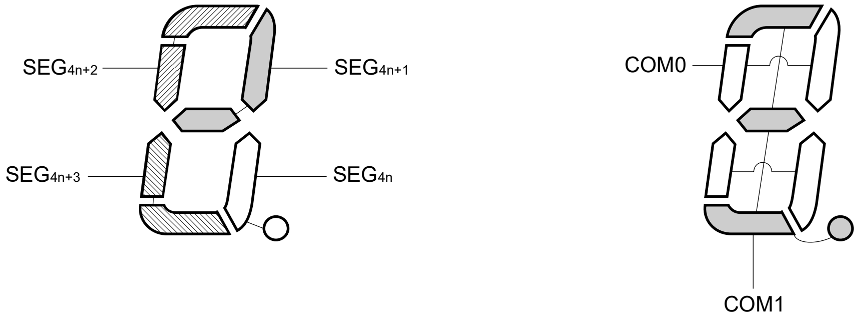
Table 13-16. Select and Deselect Voltages (COM0 and COM1)

Segment / Common	SEG12	SEG13	SEG14	SEG15
COM0	Select	Select	Deselect	Deselect
COM1	Deselect	Select	Select	Select

According to Table 13-16, it is determined that the display data register location (F040FH) that corresponds to SEG15 must contain xx10.

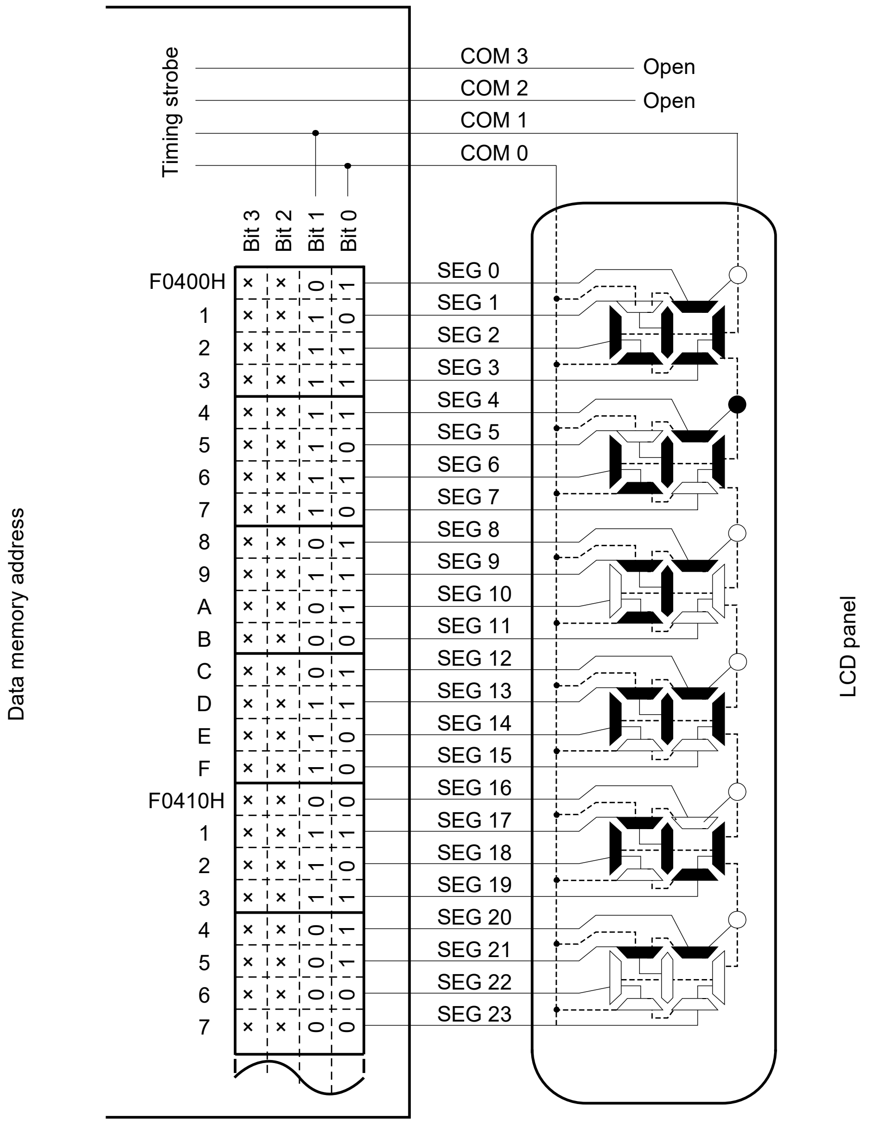
Figure 13-32 shows examples of LCD drive waveforms between the SEG15 signal and each common signal. When the select voltage is applied to SEG15 at the timing of COM1, an alternate rectangle waveform, +V_{LCD}/–V_{LCD}, is generated to turn on the corresponding LCD segment.

Figure 13-30. Two-Time-Slice LCD Display Pattern and Electrode Connections



Remark 48-pin products: n = 0 to 5
64-pin products: n = 0 to 8

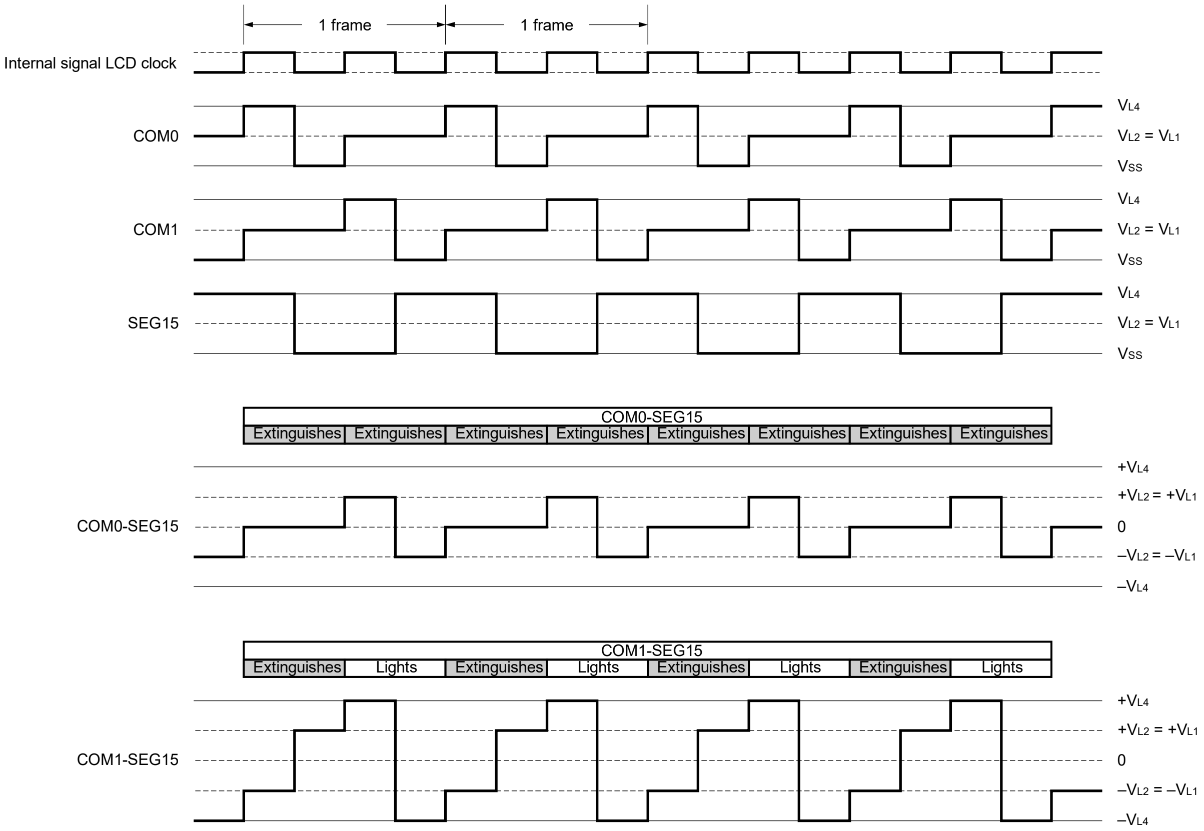
Figure 13-31. Example of Connecting Two-Time-Slice LCD Panel



Data memory address	Bit 3	Bit 2	Bit 1	Bit 0
F0400H	×	×	0	1
1	×	×	1	0
2	×	×	1	1
3	×	×	1	1
4	×	×	1	1
5	×	×	1	0
6	×	×	1	1
7	×	×	1	0
8	×	×	0	1
9	×	×	1	1
A	×	×	0	1
B	×	×	0	0
C	×	×	0	1
D	×	×	1	1
E	×	×	1	0
F	×	×	1	0
F0410H	×	×	0	0
1	×	×	1	1
2	×	×	1	0
3	×	×	1	1
4	×	×	0	1
5	×	×	0	1
6	×	×	0	0
7	×	×	0	0

Remark ×: Don’t care.

Figure 13-32. Two-Time-Slice LCD Drive Waveform Examples Between SEG15 and Each Common Signals (1/2 Bias Method)



13.10.3 Three-time-slice display example

Figure 13-34 shows how the 8-digit LCD panel having the display pattern shown in Figure 13-33 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 13-17 at the timing of the common signals COM0 to COM2; see Figure 13-33 for the relationship between the segment signals and LCD segments.

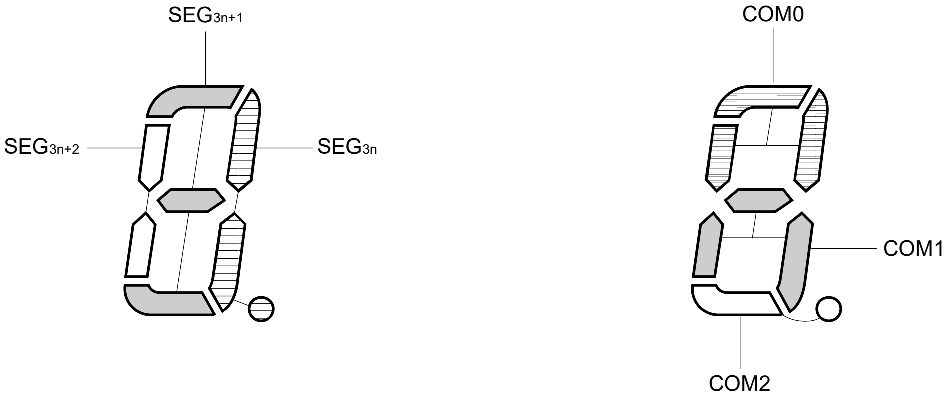
Table 13-17. Select and Deselect Voltages (COM0 to COM2)

Segment Common	SEG6	SEG7	SEG8
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	–

According to Table 13-17, it is determined that the display data register location (F0406H) that corresponds to SEG6 must contain x110.

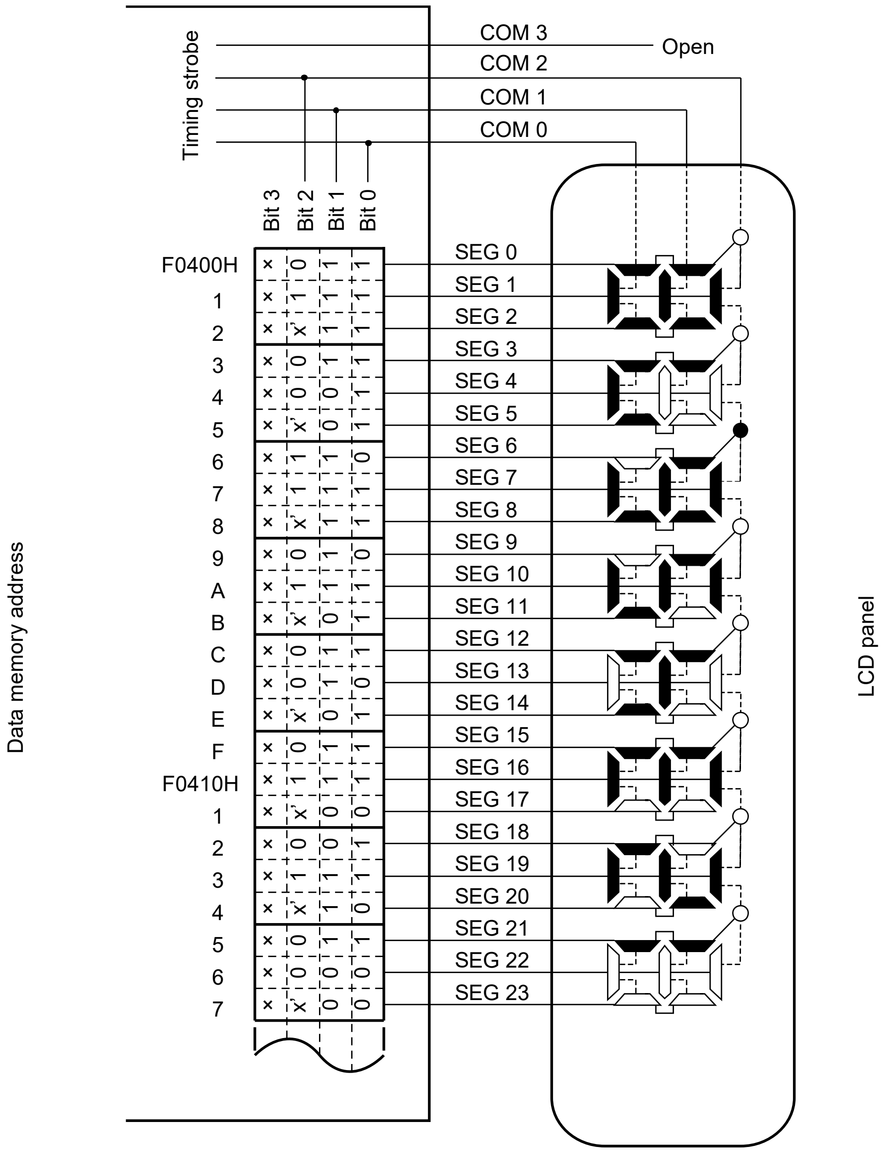
Figures 13-35 and 13-36 show examples of LCD drive waveforms between the SEG6 signal and each common signal in the 1/2 and 1/3 bias methods, respectively. When the select voltage is applied to SEG6 at the timing of COM1 or COM2, an alternate rectangle waveform, +V$_{LCD}$/–V$_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 13-33. Three-Time-Slice LCD Display Pattern and Electrode Connections



Remark 48-pin products: n = 0 to 7
64-pin products: n = 0 to 12

Figure 13-34. Example of Connecting Three-Time-Slice LCD Panel



Data memory address	Bit 3	Bit 2	Bit 1	Bit 0
F0400H	×	0	1	1
1	×	1	1	1
2	×	×'	1	1
3	×	0	1	1
4	×	0	0	1
5	×	×'	0	1
6	×	1	1	0
7	×	1	1	1
8	×	×'	1	1
9	×	0	1	0
A	×	1	1	1
B	×	×'	0	1
C	×	0	1	1
D	×	0	1	0
E	×	×'	0	1
F	×	0	1	1
F0410H	×	1	1	1
1	×	×'	0	0
2	×	0	0	1
3	×	1	1	1
4	×	×'	1	0
5	×	0	1	1
6	×	0	0	0
7	×	×'	0	0

Remark ×: Don’t care.

×’: Can be used to store any data because there is no corresponding segment in the LCD panel.

Figure 13-35. Three-Time-Slice LCD Drive Waveform Examples Between SEG6 and Each Common Signals (1/2 Bias Method)

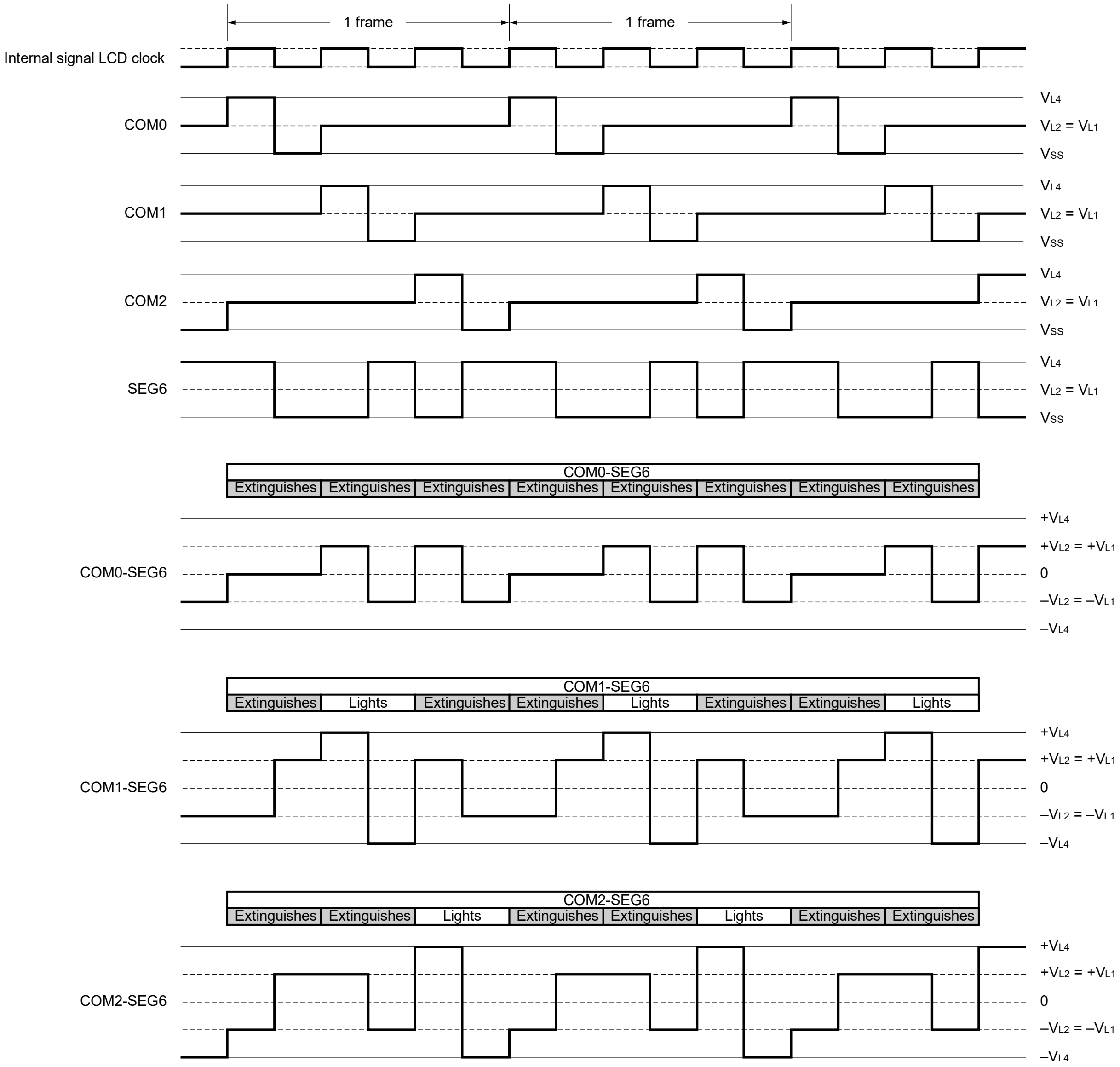
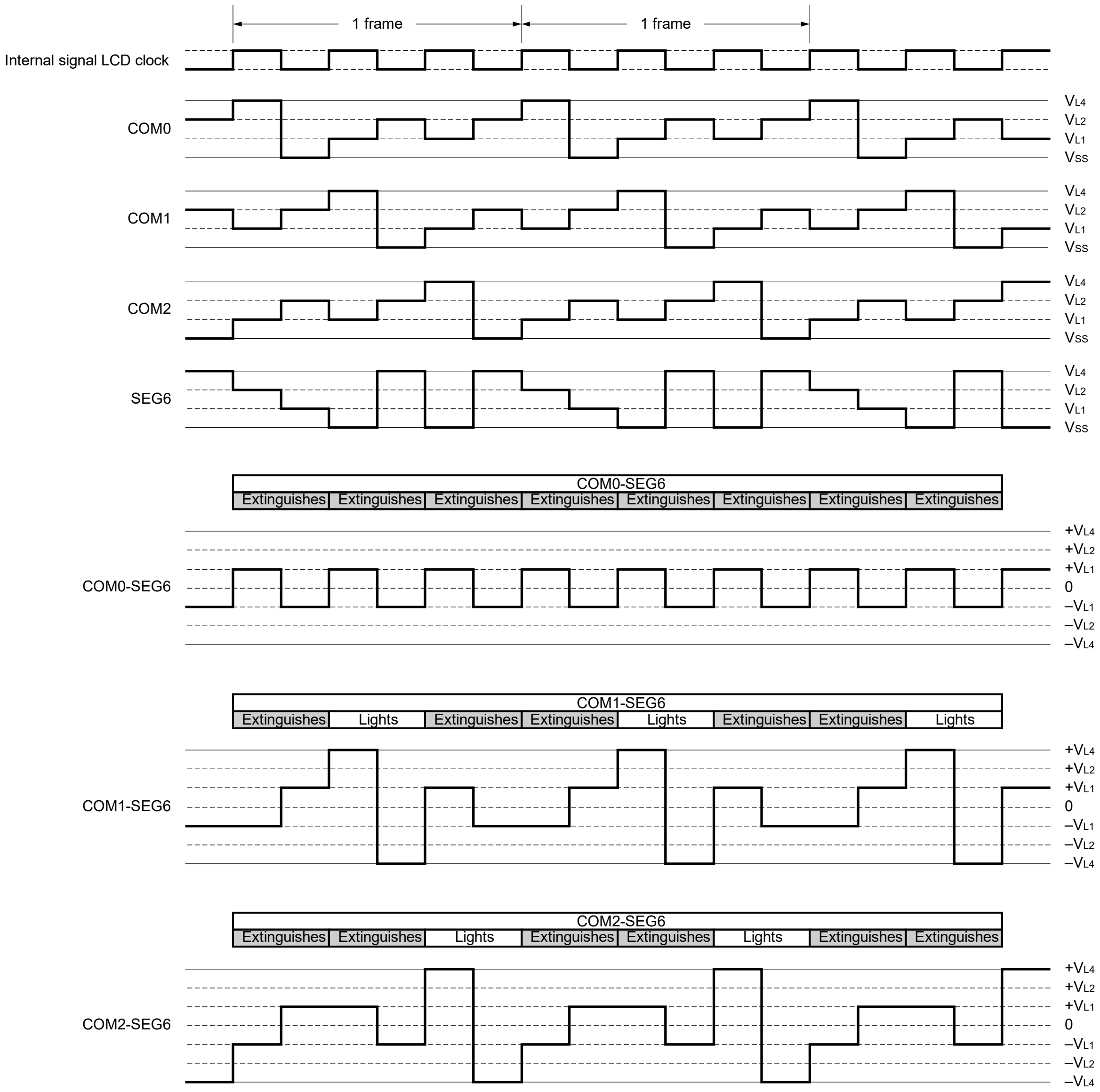


Figure 13-36. Three-Time-Slice LCD Drive Waveform Examples Between SEG6 and Each Common Signals (1/3 Bias Method)



13.10.4 Four-time-slice display example

Figure 13-38 shows how the 12-digit LCD panel having the display pattern shown in Figure 13-37 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 13-18 at the timing of the common signals COM0 to COM3; see Figure 13-37 for the relationship between the segment signals and LCD segments.

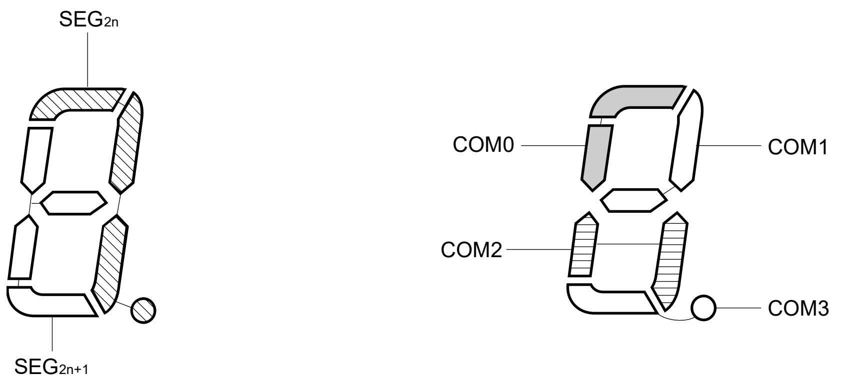
Table 13-18. Select and Deselect Voltages (COM0 to COM3)

Segment Common	SEG12	SEG13
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to Table 13-18, it is determined that the display data register location (F040CH) that corresponds to SEG12 must contain 1101.

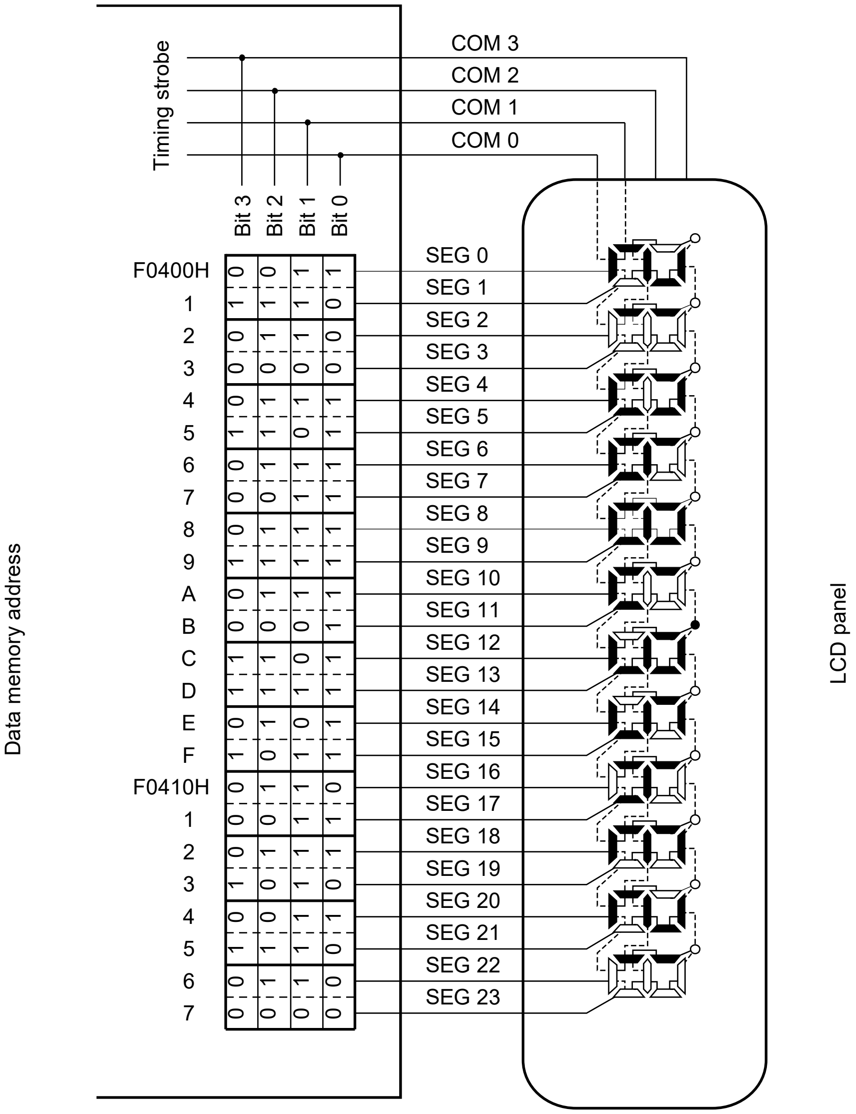
Figure 13-39 shows examples of LCD drive waveforms between the SEG12 signal and each common signal. When the select voltage is applied to SEG12 at the timing of COM0, an alternate rectangle waveform, $+V_{LCD}/-V_{LCD}$, is generated to turn on the corresponding LCD segment.

Figure 13-37. Four-Time-Slice LCD Display Pattern and Electrode Connections



Remark 48-pin products: n = 0 to 12
64-pin products: n = 0 to 18

Figure 13-38. Example of Connecting Four-Time-Slice LCD Panel



	Bit 3	Bit 2	Bit 1	Bit 0
F0400H	0	0	1	1
1	1	1	1	0
2	0	1	1	0
3	0	0	0	0
4	0	1	1	1
5	1	1	0	1
6	0	1	1	1
7	0	0	1	1
8	0	1	1	1
9	1	1	1	1
A	0	1	1	1
B	0	0	0	1
C	1	1	0	1
D	1	1	1	1
E	0	1	0	1
F	1	0	1	1
F0410H	0	1	1	0
1	0	0	1	1
2	0	1	1	1
3	1	0	1	0
4	0	0	1	1
5	1	1	1	0
6	0	1	1	0
7	0	0	0	0

Figure 13-39. Four-Time-Slice LCD Drive Waveform Examples Between SEG12 and Each Common Signals (1/3 Bias Method) (1/2)

(a) Waveform A

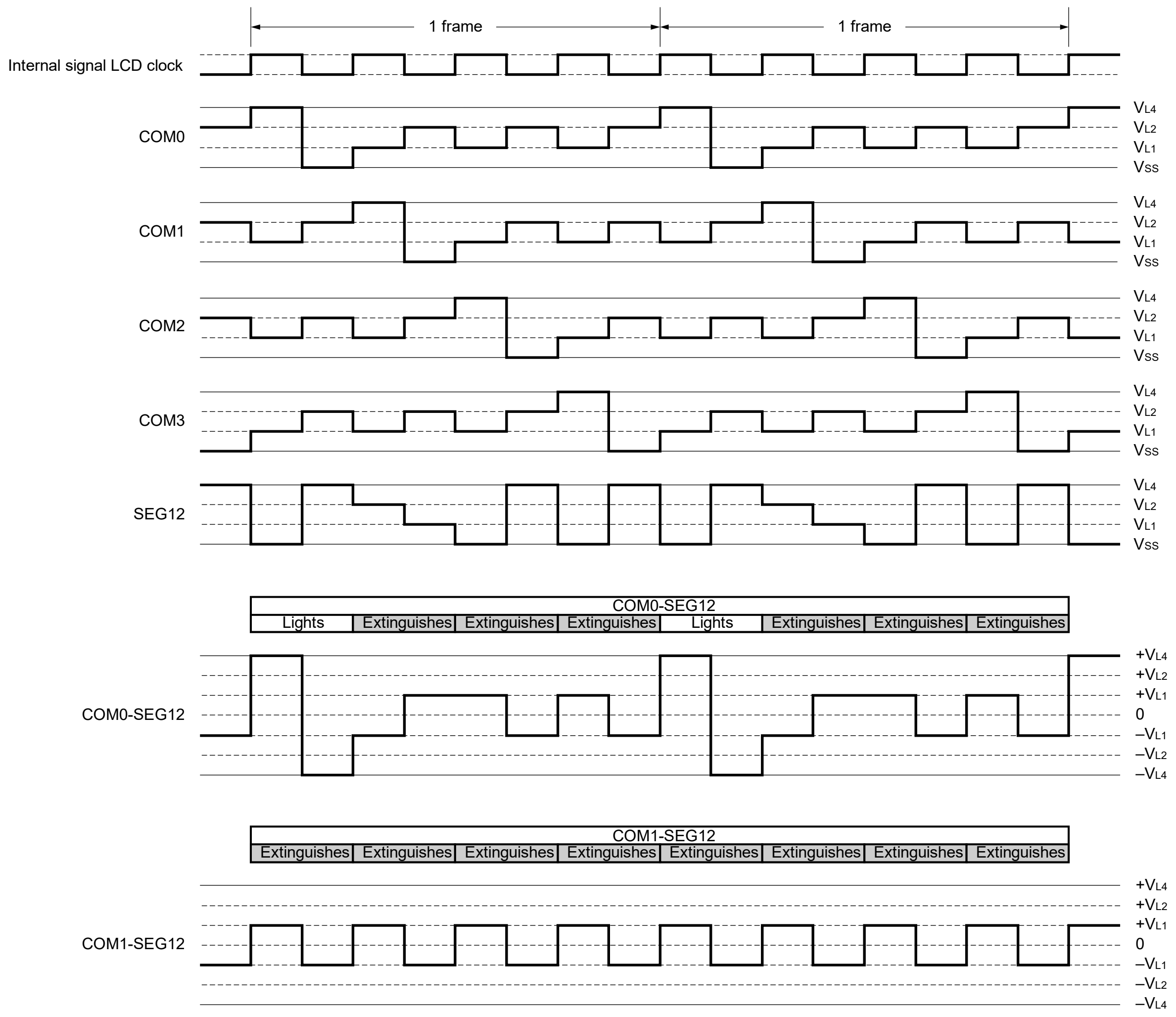
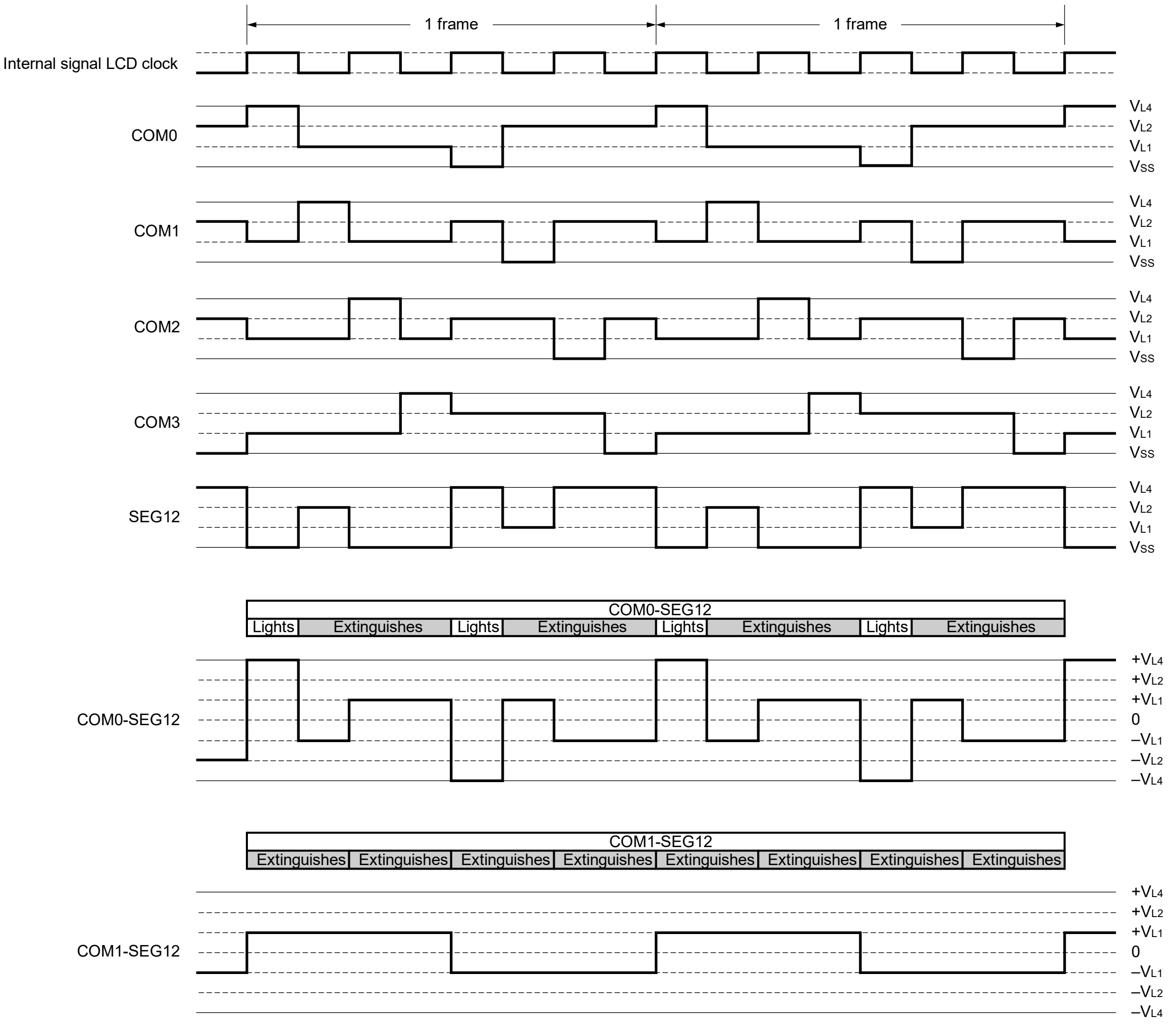


Figure 13-39. Four-Time-Slice LCD Drive Waveform Examples Between SEG12 and Each Common Signals (1/3 Bias Method) (2/2)

(b) Waveform B



13.10.5 Eight-time-slice display example

Figure 13-41 shows how the 15x8 dot LCD panel having the display pattern shown in Figure 13-40 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 13-19 at the timing of the common signals COM0 to COM7; see Figure 13-40 for the relationship between the segment signals and LCD segments.

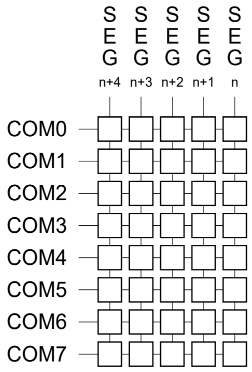
Table 13-19. Select and Deselect Voltages (COM0 to COM7)

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 13-19, it is determined that the display data register location (F0404H) that corresponds to SEG4 must contain 00110001.

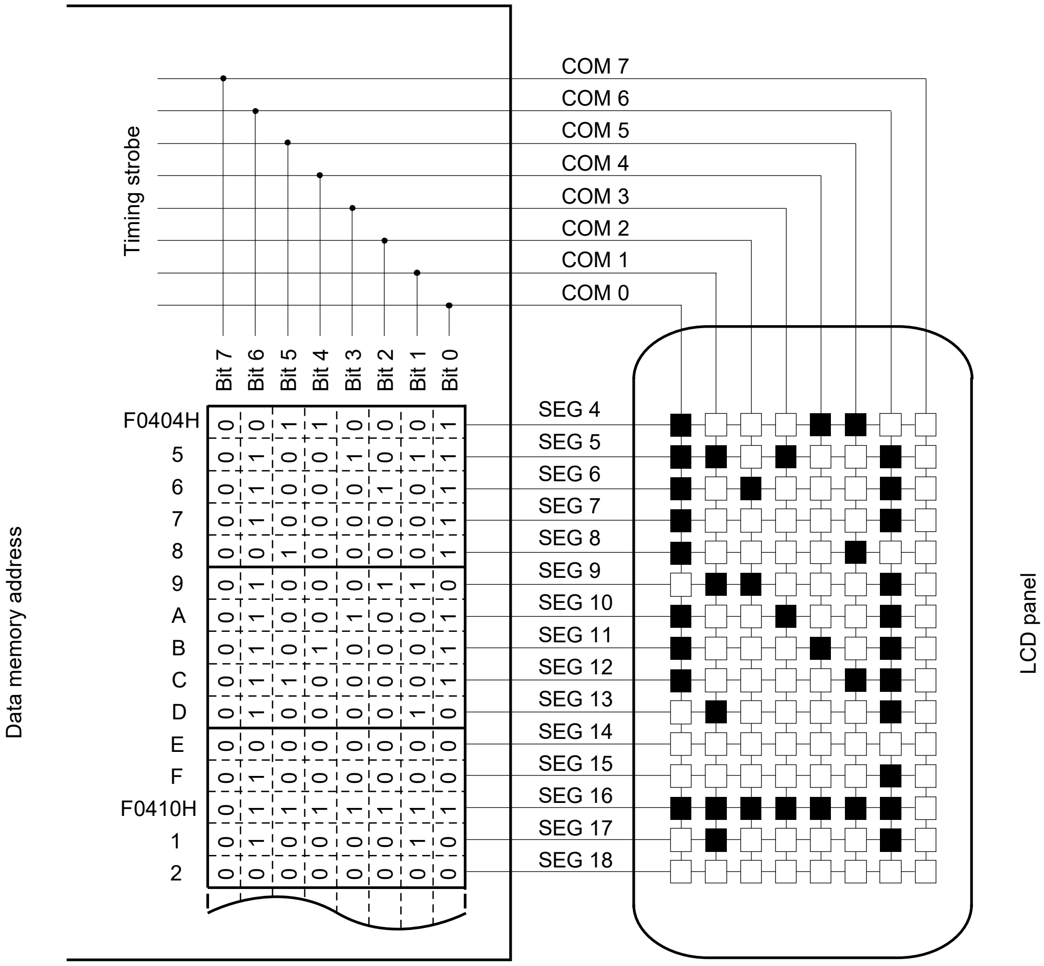
Figure 13-42 shows examples of LCD drive waveforms between the SEG4 signal and each common signal. When the select voltage is applied to SEG4 at the timing of COM0, a waveform is generated to turn on the corresponding LCD segment.

Figure 13-40. Eight-Time-Slice LCD Display Pattern and Electrode Connections



Remark 48-pin products: n = 4 to 21
64-pin products: n = 4 to 34

Figure 13-41. Example of Connecting Eight-Time-Slice LCD Panel



	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F0404H	0	0	1	1	0	0	0	1
5	0	1	0	0	1	0	1	1
6	0	1	0	0	0	1	0	1
7	0	1	0	0	0	0	0	1
8	0	0	1	0	0	0	0	1
9	0	1	0	0	0	1	1	0
A	0	1	0	0	1	0	0	1
B	0	1	0	1	0	0	0	1
C	0	1	1	0	0	0	0	1
D	0	1	0	0	0	0	1	0
E	0	0	0	0	0	0	0	0
F	0	1	0	0	0	0	0	0
F0410H	0	1	1	1	1	1	1	1
1	0	1	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0

Figure 13-42. Eight-Time-Slice LCD Drive Waveform Examples Between SEG4 and Each Common Signals (1/4 Bias Method) (1/2)

(a) Waveform A

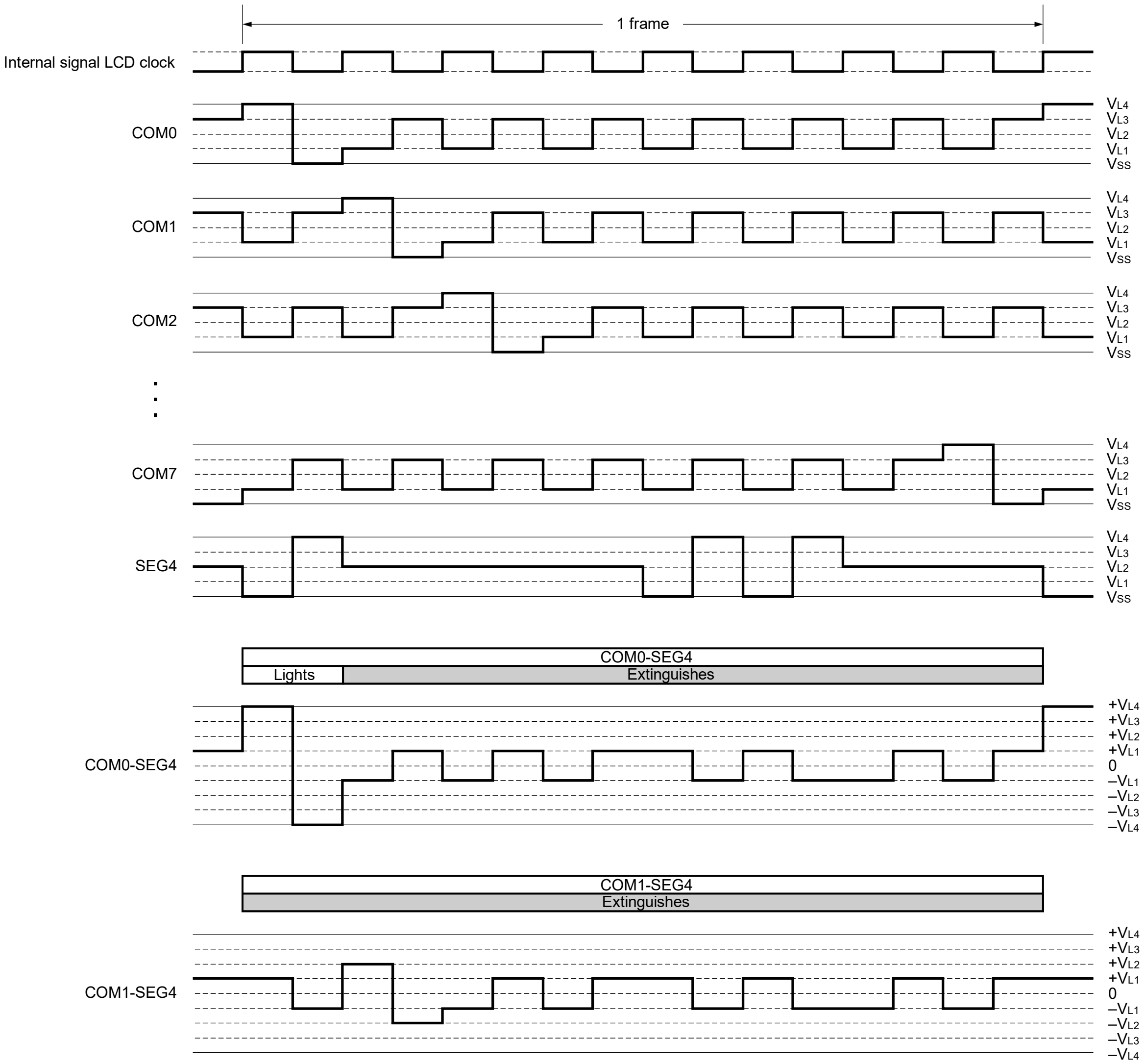
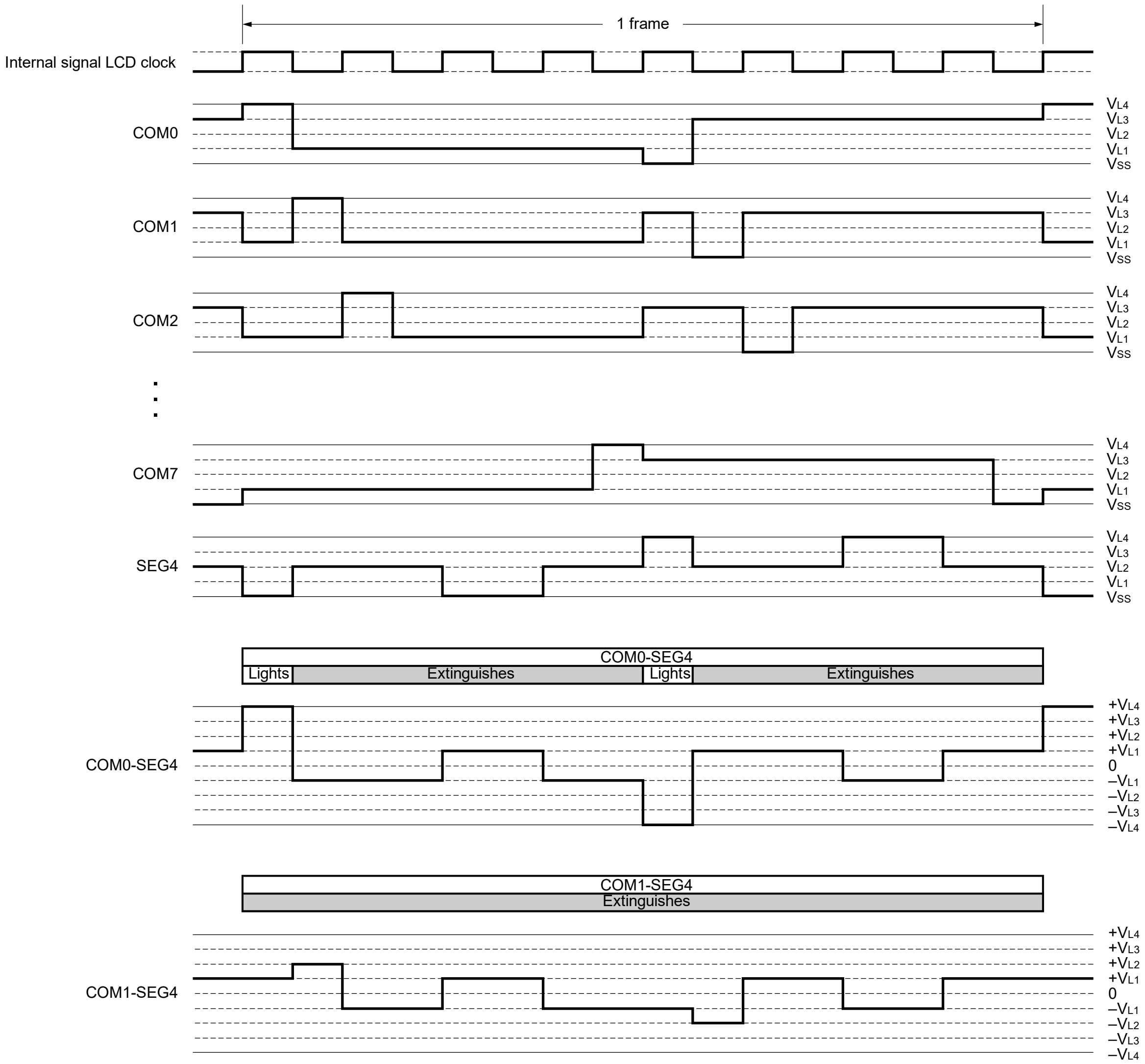


Figure 13-42. Eight-Time-Slice LCD Drive Waveform Examples Between SEG4 and Each Common Signals (1/4 Bias Method) (2/2)

(b) Waveform B



CHAPTER 14 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.

		48-pin	64-pin
Maskable interrupts	External	7	9
	Internal	23	23

14.1 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, PR10L, PR10H, PR11L, PR11H, PR12L). 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 priority of vectored interrupt servicing. For the priority order, see Table 14-1.

A standby release signal is generated and STOP, HALT, and SNOOZE modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

14.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see Table 14-1). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 14-1. Interrupt Source List (1/3)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]	64-pin	48-pin
		Name	Trigger					
Maskable	0	INTWDTI	Watchdog timer interval[Note 3] (75% of overflow time +1/2f_{IL})	Internal	00004H	(A)	√	√
	1	INTLVI	Voltage detection[Note 4]		00006H		√	√
	2	INTP0	Pin input edge detection	External	00008H	(B)	√	√
	3	INTP1			0000AH		√	√
	4	INTP2			0000CH		√	√
	5	INTP3			0000EH		√	√
	6	INTP4			00010H		√	√
	7	INTP5			00012H		√	√
	8	INTST0	UART0 transmission transfer end or buffer empty interrupt	Internal	00018H	(A)	√	√
		INTCSI00	CSI00 transfer end or buffer empty interrupt				√	√
	9	INTSR0	UART0 reception transfer end		0001AH		√	√
		INTCSI01	CSI01 transfer end or buffer empty interrupt				√	√
	10	INTSRE0	UART0 reception communication error occurrence		0001CH		√	√
		INTTM01H	End of timer channel 01 count or capture (at higher 8-bit timer operation)				√	√
	11	INTTM00	End of timer channel 00 count or capture		00020H		√	√
	12	INTTM03H	End of timer channel 03 count or capture (at higher 8-bit timer operation)		00024H		√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 24 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 14-1.

3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

Remark √: Mounted

–: Not mounted

Table 14-1. Interrupt Source List (2/3)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]	64-pin	48-pin
		Name	Trigger					
Maskable	13	INTTM01	End of timer channel 01 count or capture	Internal	00028H	(A)	√	√
	14	INTTM02	End of timer channel 02 count or capture		0002AH		√	√
	15	INTTM03	End of timer channel 03 count or capture		0002CH		√	√
	16	INTAD	End of A/D conversion		0002EH		√	√
	17	INTRTC	Fixed-cycle signal of real-time clock/alarm match detection		00030H		√	√
	18	INTIT	Interval signal detection		00032H		√	√
	19	INTKR	Key return signal detection	External	00034H	(C)	√	√
	20	INTTM06	End of timer channel 06 count or capture	Internal	00040H	(A)	√	√
	21	INTTM07	End of timer channel 07 count or capture		00042H		√	√
	22	INTP6	Pin input edge detection	External	00046H	(B)	√	–
	23	INTP7			00048H		√	–
	24	INTFL	End of sequencer interrupt[Note 3]	Internal	0004CH	(A)	√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 24 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 14-1.

3. Be used at the flash self programming library or the data flash library.

Remark √: Mounted

–: Not mounted

Table 14-1. Interrupt Source List (3/3)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]	64-pin	48-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				√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 24 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 14-1.

3. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

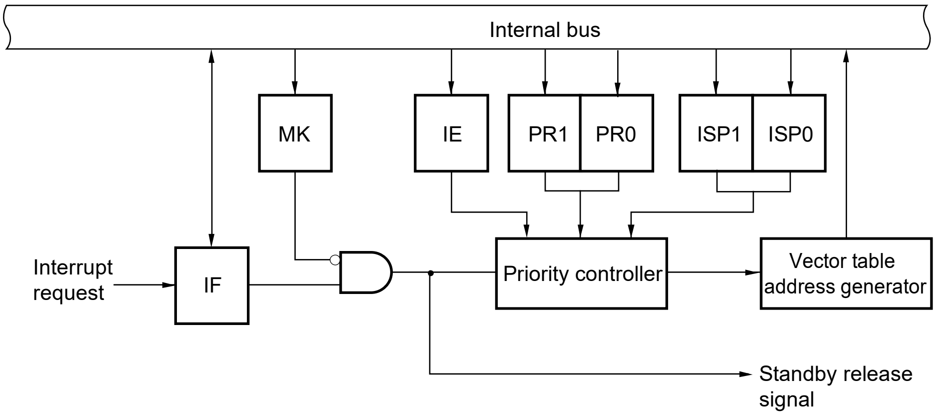
4. When the instruction code in FFH is executed.

Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

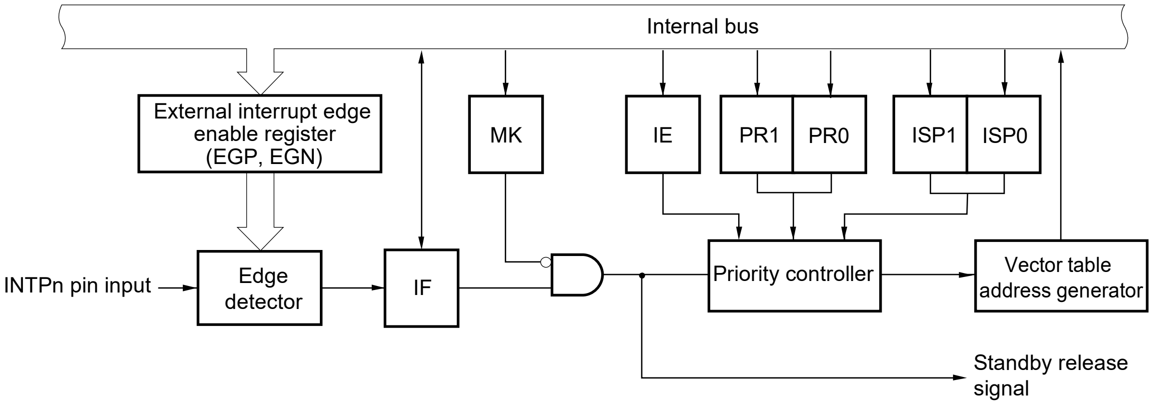
Remark √: Mounted

Figure 14-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)



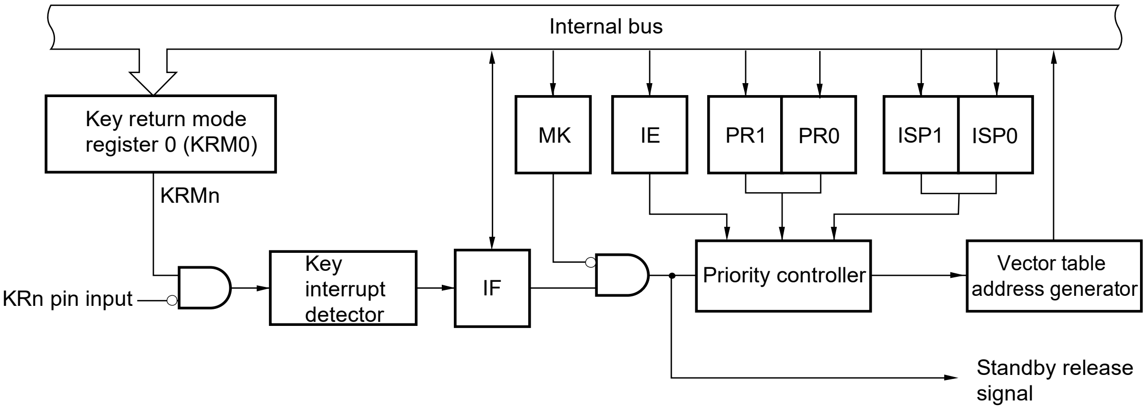
IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP0:	In-service priority flag 0
ISP1:	In-service priority flag 1
MK:	Interrupt mask flag
PR0:	Priority specification flag 0
PR1:	Priority specification flag 1

Remark 48-pin: n = 0 to 5

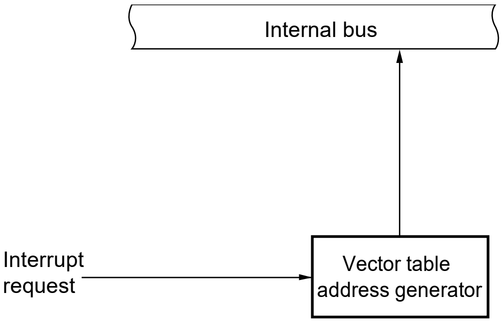
64-pin: n = 0 to 7

Figure 14-1. Basic Configuration of Interrupt Function (2/2)

(C) External maskable interrupt (INTKR)



(D) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

Remark n = 0 to 3

14.3 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)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L)
- External interrupt rising edge enable register (EGP0)
- External interrupt falling edge enable register (EGN0)
- Program status word (PSW)

Table 14-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 14-2. Flags Corresponding to Interrupt Request Sources (1/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		64-pin	48-pin
		Register		Register		Register		
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	√	√
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1		√	√
INTP0	PIF0		PMK0		PPR00, PPR10		√	√
INTP1	PIF1		PMK1		PPR01, PPR11		√	√
INTP2	PIF2		PMK2		PPR02, PPR12		√	√
INTP3	PIF3		PMK3		PPR03, PPR13		√	√
INTP4	PIF4		PMK4		PPR04, PPR14		√	√
INTP5	PIF5		PMK5		PPR05, PPR15		√	√

Remark √: Mounted
–: Not mounted

Table 14-2. Flags Corresponding to Interrupt Request Sources (2/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		64-pin	48-pin
		Register		Register		Register		
INTST0[Note 1]	STIF0[Note 1]	IF0H	STMK0[Note 1]	MK0H	STPR00, STPR10[Note 1]	PR00H, PR10H	√	√
INTCSI00[Note 1]	CSIIF00[Note 1]		CSIMK00[Note 1]		CSIPR000, CSIPR100[Note 1]		√	√
INTSR0[Note 2]	SRIF0[Note 2]		SRMK0[Note 2]		SRPR00, SRPR10[Note 2]		√	√
INTCSI01[Note 2]	CSIIF01[Note 2]		CSIMK01[Note 2]		CSIPR001, CSIPR101[Note 2]		√	√
INTSRE0[Note 3]	SREIF0[Note 3]		SREMK0[Note 3]		SREPR00, SREPR10[Note 3]		√	√
INTTM01H[Note 3]	TMIF01H[Note 3]		TMMK01H[Note 3]		TMPR001H, TMPR101H[Note 3]		√	√
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		√	√
INTTM03H	TMIF03H	IF1L	TMMK03H	MK1L	TMPR003H, TMPR103H	PR01L, PR11L	√	√
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		√	√
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		√	√
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		√	√
INTAD	ADIF		ADMK		ADPR0, ADPR1		√	√
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1		√	√
INTIT	ITIF		ITMK		ITPR0, ITPR1		√	√

Notes 1. If one of the interrupt sources INTST0 and INTCSI00 is generated, bit 2 of the IF0H register is set to 1. Bit 2 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

2. If one of the interrupt sources INTSR0 and INTCSI01 is generated, bit 3 of the IF0H register is set to 1. Bit 3 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

3. Do not use a UART0 reception error interrupt and an interrupt of channel 1 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART0 reception error interrupt is not used (EOC01 = 0), UART0 and channel 1 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If one of the interrupt sources INTSRE0 and INTTM01H is generated, bit 7 of the IF0H register is set to 1. Bit 7 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

Remark √: Mounted

Table 14-2. Flags Corresponding to Interrupt Request Sources (3/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		64-pin	48-pin
		Register		Register		Register		
INTKR	KRIF	IF1H	KRMK	MK1H	KRPR0, KRPR1	PR01H, PR11H	√	√
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106		√	√
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		√	√
INTP6	PIF6	IF2L	PMK6	MK2L	PPR06, PPR16	PR02L, PR12L	√	–
INTP7	PIF7		PMK7		PPR07, PPR17		√	–
INTFL	FLIF		FLMK		FLPR0, FLPR1		√	√

Remark √: Mounted

–: Not mounted

14.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L)

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, and IF2L 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 are combined to form 16-bit registers IF0 and IF1, 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 14-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L) (64-pin products)

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	0	TMIF00	0	SREIF0 TMIF01H	SRIF0 CSIIF01	STIF0 CSIIF00	0	0

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
IF1L	ITIF	RTCIF	ADIF	TMIF03	TMIF02	TMIF01	0	TMIF03H

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	5	4	3	2	1	<0>
IF1H	TMIF07	TMIF06	0	0	0	0	0	KRIF

Address: FFFD0H After reset: 00H R/W

Symbol	7	6	5	<4>	3	<2>	<1>	0
IF2L	0	0	0	FLIF	0	PIF7	PIF6	0

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

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

Cautions 1. The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 14-2. Be sure to set bits that are not available to the initial value.

2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as “IF0L.0 = 0;” or “_asm(“clr1 IF0L, 0”);” because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as “IF0L &= 0xfe;” and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of the another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between “mov a, IF0L” and “mov IF0L, a”, the flag is cleared to 0 at “mov IF0L, a”. Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

14.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

The MK0L, MK0H, MK1L, MK1H, and MK2L registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the MK0L and MK0H registers, and the MK1L and MK1H registers are combined to form 16-bit registers MK0 and MK1, 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 14-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L)(64-pin products)

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	1	TMMK00	1	SREMK0 TMMK01H	SRMK0 CSIMK01	STMK0 CSIMK00	1	1

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
MK1L	ITMK	RTCMK	ADMK	TMMK03	TMMK02	TMMK01	1	TMMK03H

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	5	4	3	2	1	<0>
MK1H	TMMK07	TMMK06	1	1	1	1	1	KRMK

Address: FFFD4H After reset: FFH R/W

Symbol	7	6	5	<4>	3	<2>	<1>	0
MK2L	1	1	1	FLMK	1	PMK7	PMK6	1

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution The above is the bit layout for the 64-pin products. The available bits differ depending on the product. For details about the bits available for each product, see Table 14-2. Be sure to set bits that are not available to the initial value.

14.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L)

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, or 2L).

The PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, and PR12L 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 PR10L and PR10H registers, and the PR11L and PR11H registers are combined to form 16-bit registers PR00, PR01, PR10, and PR11, 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 14-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L) (64-pin products) (1/2)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	7	<6>	5	<4>	<3>	<2>	1	0
PR00H	1	TMPR000	1	SREPR00 TMPR001H	SRPR00 CSIPR001	STPR00 CSIPR000	1	1

Address: FFFEDH After reset: FFH R/W

Symbol	7	<6>	5	<4>	<3>	<2>	1	0
PR10H	1	TMPR100	1	SREPR10 TMPR101H	SRPR10 CSIPR101	STPR10 CSIPR100	1	1

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PR01L	ITPR0	RTCPR0	ADPR0	TMPR003	TMPR002	TMPR001	1	TMPR003H

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PR11L	ITPR1	RTCPR1	ADPR1	TMPR103	TMPR102	TMPR101	1	TMPR103H

Figure 14-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L) (64-pin products) (2/2)

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	5	4	3	2	1	<0>
PR01H	TMPR007	TMPR006	1	1	1	1	1	KRPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	5	4	3	2	1	<0>
PR11H	TMPR107	TMPR106	1	1	1	1	1	KRPR1

Address: FFFD8H After reset: FFH R/W

Symbol	7	6	5	<4>	3	<2>	<1>	0
PR02L	1	1	1	FLPR0	1	PPR07	PPR06	1

Address: FFFDCH After reset: FFH R/W

Symbol	7	6	5	<4>	3	<2>	<1>	0
PR12L	1	1	1	FLPR1	1	PPR17	PPR16	1

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 above is the bit layout for the 64-pin products. The available bits differ depending on the product. For details about the bits available for each product, see Table 14-2. Be sure to set bits that are not available to the initial value.

14.3.4 External interrupt rising edge enable register (EGP0), external interrupt falling edge enable register (EGN0)

These registers specify the valid edge for INTP0 to 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 14-5. Format of External Interrupt Rising Edge Enable Register (EGP0) and External Interrupt Falling Edge Enable Register (EGN0) (64-pin products)

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 14-3 shows the ports corresponding to the EGPn and EGNn bits.

Table 14-3. Ports Corresponding to EGPn and EGNn Bits

Detection Enable Bit		Edge Detection Port	Interrupt Request Signal	64-pin	48-pin
EGP0	EGN0	P137	INTP0	√	√
EGP1	EGN1	P15	INTP1	√	√
EGP2	EGN2	P16	INTP2	√	√
EGP3	EGN3	P31	INTP3	√	√
EGP4	EGN4	P32	INTP4	√	√
EGP5	EGN5	P50	INTP5	√	√
EGP6	EGN6	P52	INTP6	√	–
EGP7	EGN7	P43	INTP7	√	–

Caution Select the port mode by clearing the EGPn and EGNn bits to 0 because an edge may be detected when the external interrupt function is switched to the port function.

Remarks 1. n = 0 to 7

2. √: Mounted

–: Not mounted

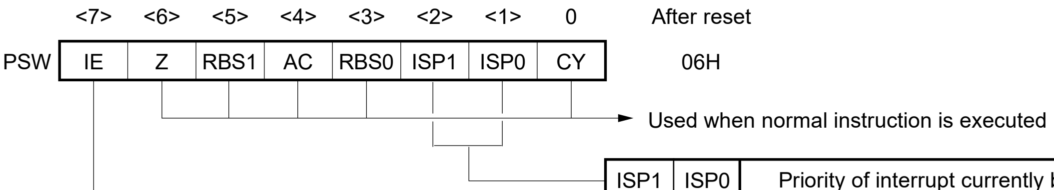
14.3.5 Program status word (PSW)

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 14-6. 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

14.4 Interrupt Servicing Operations

14.4.1 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 14-4 below.

For the interrupt request acknowledgment timing, see Figures 14-8 and 14-9.

Table 14-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time^{Note}
Servicing time	9 clocks	16 clocks

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

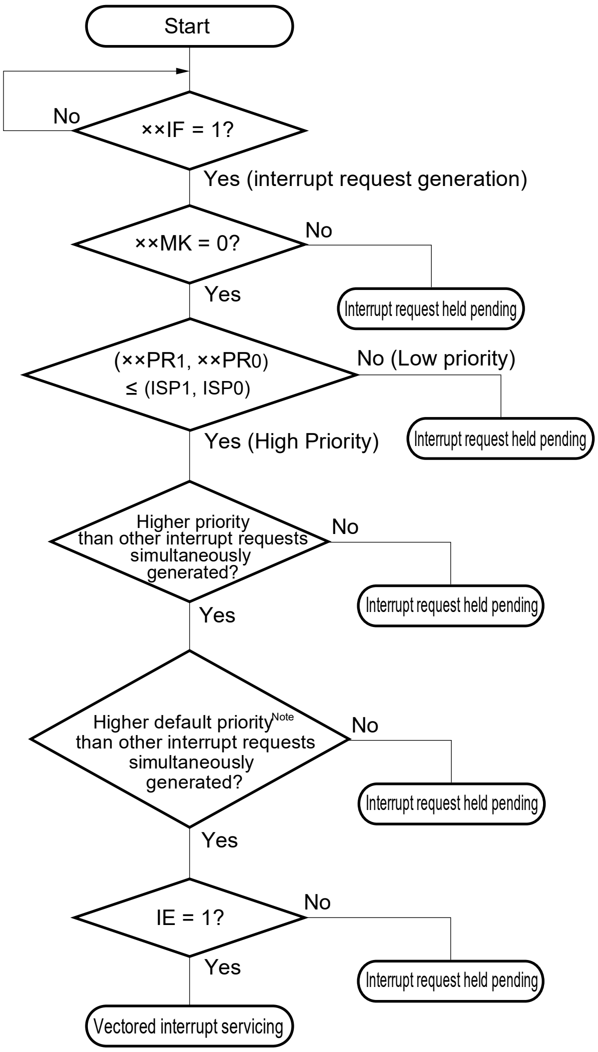
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 14-7 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (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.

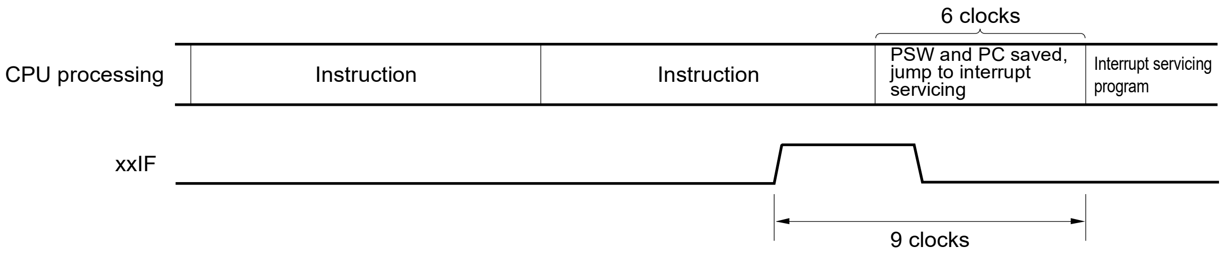
Figure 14-7. Interrupt Request Acknowledgment Processing Algorithm



××IF:	Interrupt request flag
××MK:	Interrupt mask flag
××PR0:	Priority specification flag 0
××PR1:	Priority specification flag 1
IE:	Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1:	Flag that indicates the priority level of the interrupt currently being serviced (see Figure 14-6)

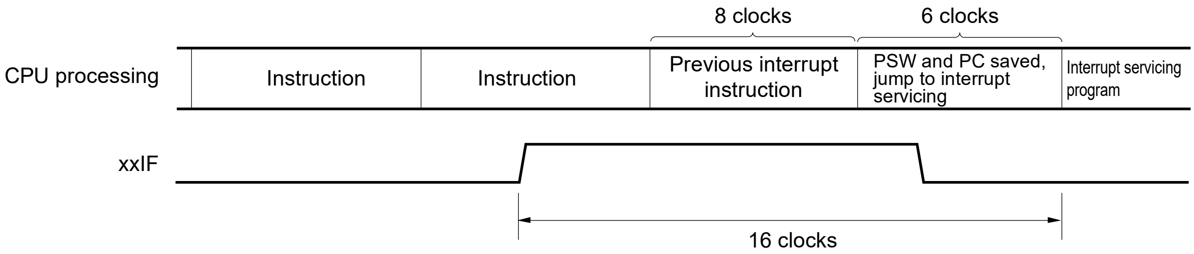
Note For the default priority, refer to **Table 14-1 Interrupt Source List**.

Figure 14-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

Figure 14-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

14.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (0007EH, 0007FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

Caution Can not use the RETI instruction for restoring from the software interrupt.

14.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 14-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 14-10 shows multiple interrupt servicing examples.

Table 14-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0 ISP0 = 0	O	×	×	×	×	×	×	×	O
	ISP1 = 0 ISP0 = 1	O	×	O	×	×	×	×	×	O
	ISP1 = 1 ISP0 = 0	O	×	O	×	O	×	×	×	O
	ISP1 = 1 ISP0 = 1	O	×	O	×	O	×	O	×	O
Software interrupt		O	×	O	×	O	×	O	×	O

Remarks 1. O: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

3. ISP0, ISP1, and IE are flags contained in the PSW.

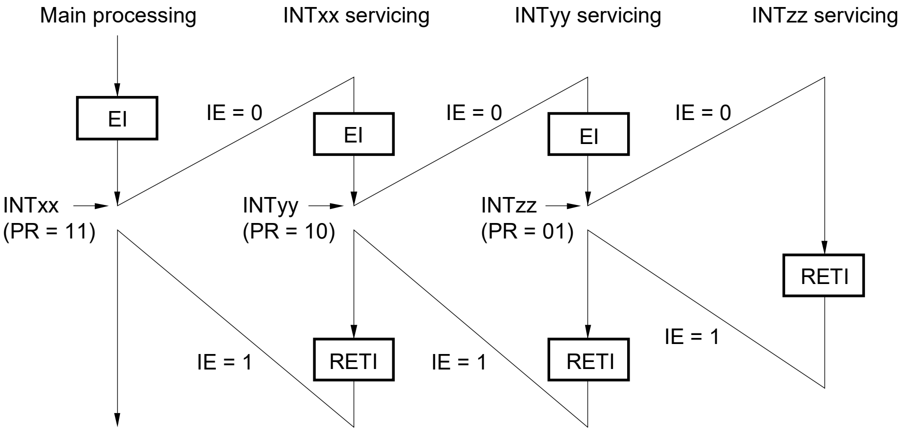
ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.
ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.
ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.
ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment (all interrupts are enabled).
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

4. PR is a flag contained in the PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, and PR12L 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)

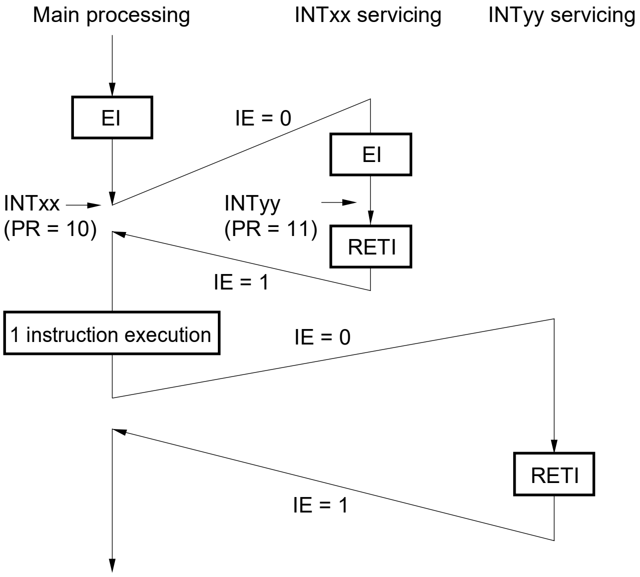
Figure 14-10. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

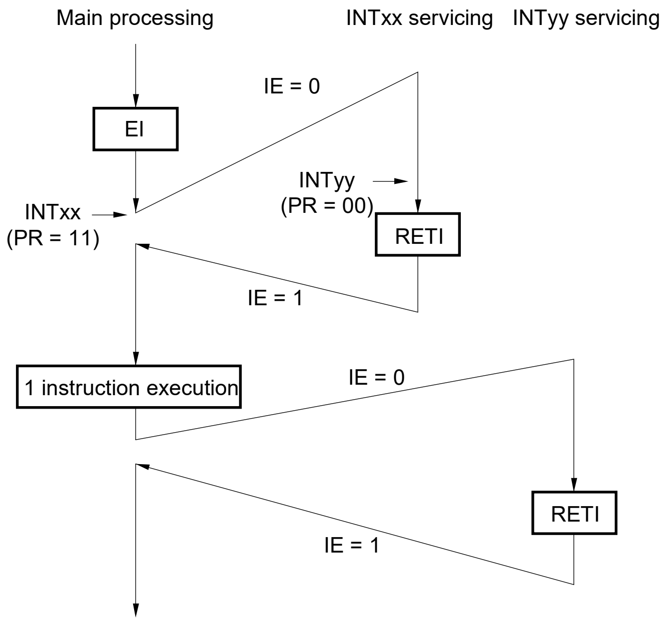


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 14-10. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

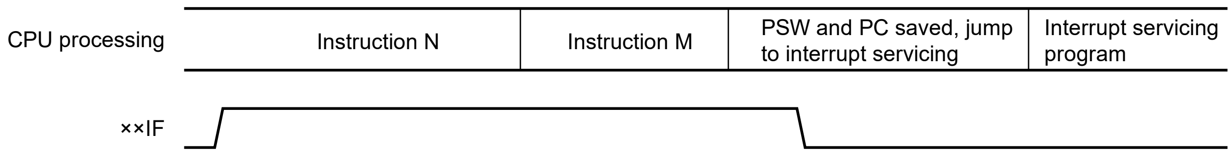
14.4.4 Interrupt request hold

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- Manipulation instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, MK0L, MK0H, MK1L, MK1H, MK2L, PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, and PR12L registers

Figure 14-11 shows the timing at which interrupt requests are held pending.

Figure 14-11. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 15 KEY INTERRUPT FUNCTION

15.1 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 KR3).

Table 15-1. Assignment of Key Interrupt Detection Pins

Key interrupt input pins	Key return mode registers (KRM)
KR0	KRM0
KR1	KRM1
KR2	KRM2
KR3	KRM3

15.2 Configuration of Key Interrupt

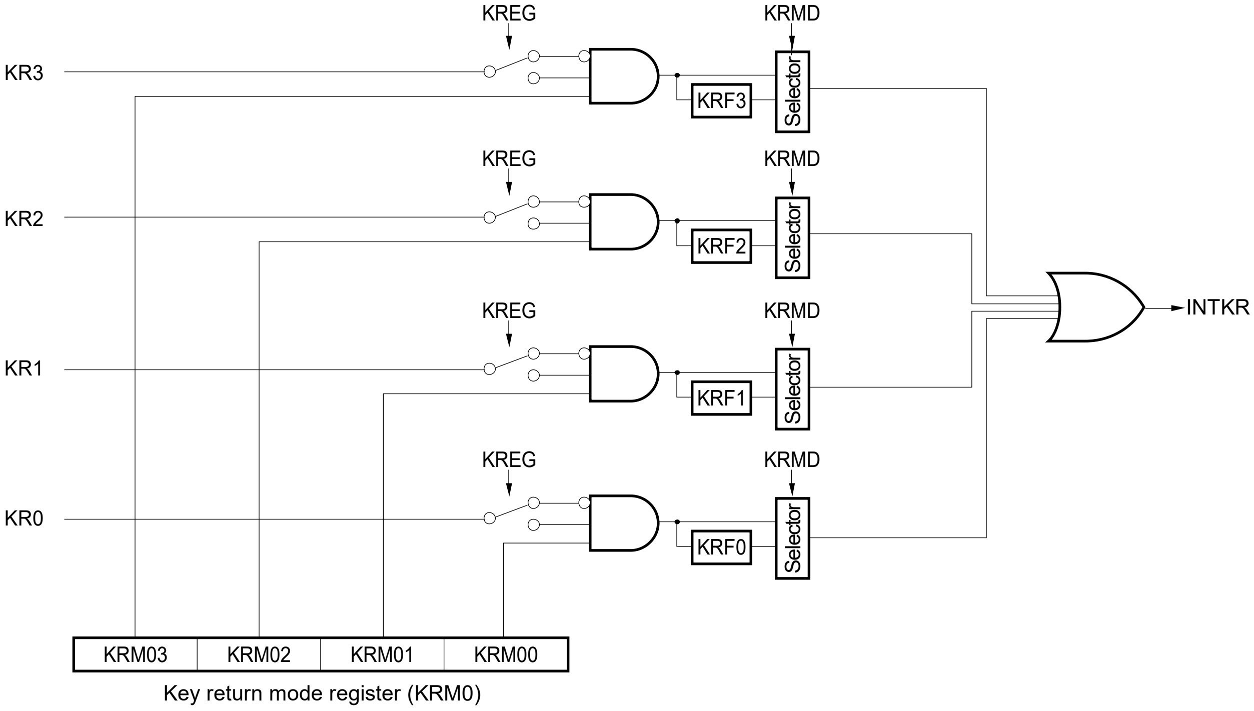
The key interrupt includes the following hardware.

Table 15-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return control register (KRCTL) Key return mode register 0 (KRM0) Port mode registers3, 7 (PM3, PM7) Note

Note The port mode registers (PMxx) to be set differ depending on the product. For details, **see 15.3.4 Port mode registers 3, 7 (PM3, PM7).**

Figure 15-1. Block Diagram of Key Interrupt



15.3 Register Controlling Key Interrupt

The key interrupt function is controlled by the following five registers:

- Key return control register (KRCTL)
- Key return mode register 0 (KRM0)
- Key return flag register (KRF)
- Port mode registers 3, 7 (PM3, PM7) **Note**

Note The port mode registers (PMxx) to be set differ depending on the product. For details, see **15.3.4 Port mode registers 3, 7 (PM3, PM7).**

15.3.1 Key return control register (KRCTL)

This register controls the usage of the key interrupt flags (KRF0 to KRF3) 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 15-2. 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 t key interrupt flags (KRF0 to KRF3)
0	Does not use key interrupt flags
1	Uses key interrupt flags

KRMD	Selection of Detection Edge (KR0 to KR3)
0	Falling edge
1	Rising edge

15.3.2 Key return mode register 0 (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 15-3. Format of Key Return Mode Register 0 (KRM0)

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM0	0	0	0	0	KRM03	KRM02	KRM01	KRM00

KRM0n	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Cautions 1. The internal pull-up resistor can be used by setting the corresponding bits to 1 in the pull-up resistor registers 3, 7 (PU3, PU7) of key interrupt input pins.

2. An interrupt will be generated if the target bit of the KRM0 register is set while a low level (when KREG = 0)/high level (when KREG = 1) is being input to the key interrupt input pin. To ignore this interrupt, set the KRM0 register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input high-level width/low-level width (see 27.4 AC Characteristics).

3. The pins not used in the key interrupt mode can be used as normal ports.

Remark n = 0 to 3

15.3.3 Key return flag register (KRF)

This register controls the key interrupt flags (KRF0 to KRF3).

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

Reset signal generation clears this register to 00H.

Figure 15-4. Format of Key return Flag Register (KRF)

Address: FFF35H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRF	0	0	0	0	KRF3	KRF2	KRF1	KRF0

KRFn	Key interrupt flag (n = 0 to 3)
0	No key interrupt signal has been detected.
1	A key interrupt signal has been detected.

Caution When KRMD = 0, setting the KRFn bit to 1 is prohibited.

15.3.4 Port mode registers 3, 7 (PM3, PM7)

These registers set the input and output of port 3, 7 in 1-bit units.

The presence or absence of key input pins depends on the product. When using the key interrupt function, set the following port mode registers according to the product used.

48-pin products:	PM3, PM7
64-pin products:	PM7

When using P30/KR3 to P32/KR1, or P70/KR0 to P73/KR3 as a key input pin, set the bit of the port mode registers (PM3, PM7) corresponding to the port pin to 1.

The PM3, PM7 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to FFH.

Figure 15-5. Format of Port Mode Register 7 (PM7) (64-pin products)

Address: FFF27H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PM7	1	1	1	PM74	PM73	PM72	PM71	PM70

PM7n	I/O mode selection for P7n pin (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark The figure shown above presents the format of port mode register 7 of the 64-pin products. The format of the port mode register of other products, see **Table 4-3 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product**.

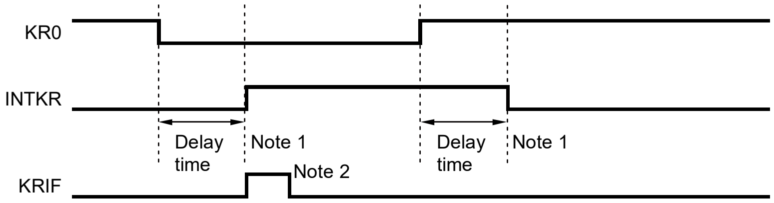
15.4 Key Interrupt Operation

15.4.1 When not using the key interrupt flag (KRMD = 0)

A key interrupt (INTKR) is generated when the valid edge specified by the setting of the KREG bit is input to a key interrupt pin (KR0 to KR3). The channel to which the valid edge was input can be identified by reading the port register and checking the port level after the key interrupt (INTKR) is generated.

The INTKR signal changes according to the input level of the key interrupt input pin (KR0 to KR3).

Figure 15-6. Operation of INTKR Signal When a Key Interrupt is Input to a Single Channel (When KRMD = 0 and KREG = 0)

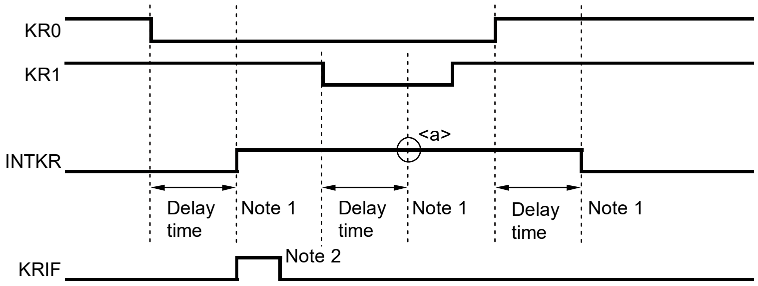


Notes 1. The maximum delay time is the maximum value of the high-level width and low-level width of the key interrupt input (see **27.4 AC Characteristics** for details).

2. Acknowledgment of vectored interrupt request or bit cleared by software

The operation when a valid edge is input to multiple key interrupt input pins is shown in Figure 15-7 below. The INTKR signal is set while a low level is being input to one pin (when KREG is set to 0). Therefore, even if a falling edge is input to another pin in this period, a key interrupt (INTKR) will not be generated again (<1> in the figure).

Figure 15-7. Operation of INTKR Signal When Key Interrupts Are Input to Multiple Channels (When KRMD = 0 and KREG = 0)



Notes 1. The maximum delay time is the maximum value of the high-level width and low-level width of the key interrupt input (see **27.4 AC Characteristics** for details).

2. Acknowledgment of vectored interrupt request or bit cleared by software

15.4.2 When using the key interrupt flag (KRMD = 1)

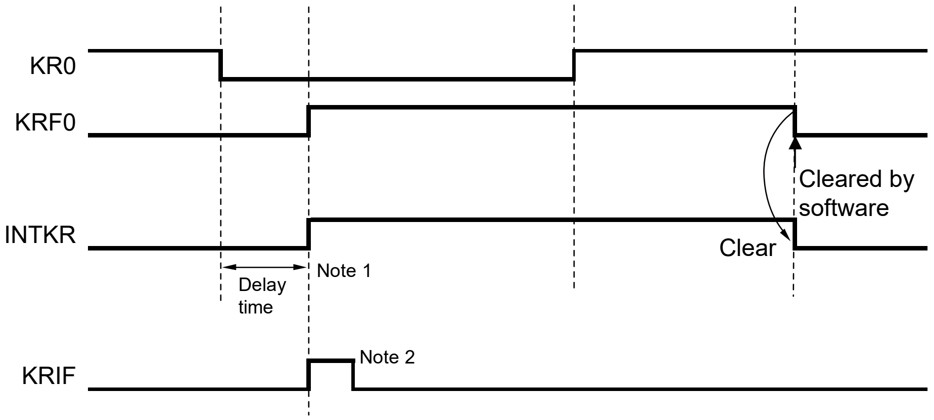
A key interrupt (INTKR) is generated when the valid edge specified by the setting of the KREG bit is input to a key interrupt pin (KR0 to KR3). The channels to which the valid edge was input can be identified by reading the key return flag register (KRF) after the key interrupt (INTKR) is generated.

If the KRMD bit is set to 1, the INTKR signal is cleared by clearing the corresponding bit in the KRF register.

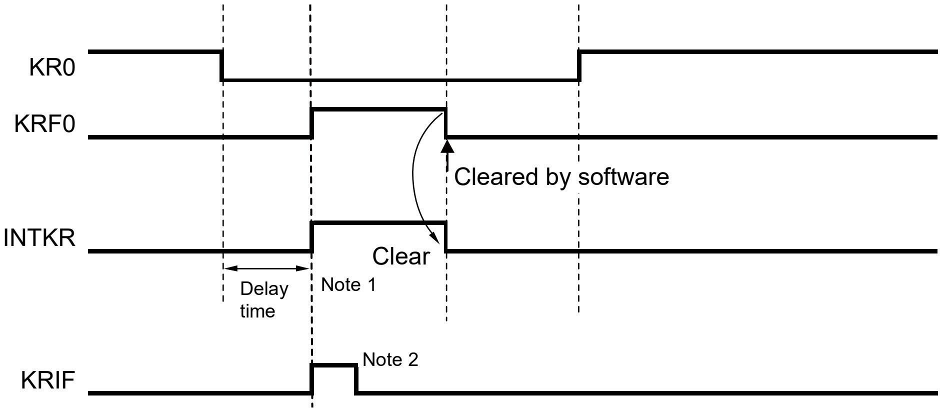
As shown in Figure 15-8, only one interrupt is generated each time a falling edge is input to one channel (when KREG = 0), regardless of whether the KRFn bit is cleared before or after a rising edge is input.

Figure 15-8. Basic Operation of the INTKR Signal When the Key Interrupt Flag Is Used (When KRMD = 1 and KREG = 0)

(a) When KRF0 is cleared after a rising edge is input to the KR0 pin



(b) When KRF0 is cleared before a rising edge is input to the KR0 pin

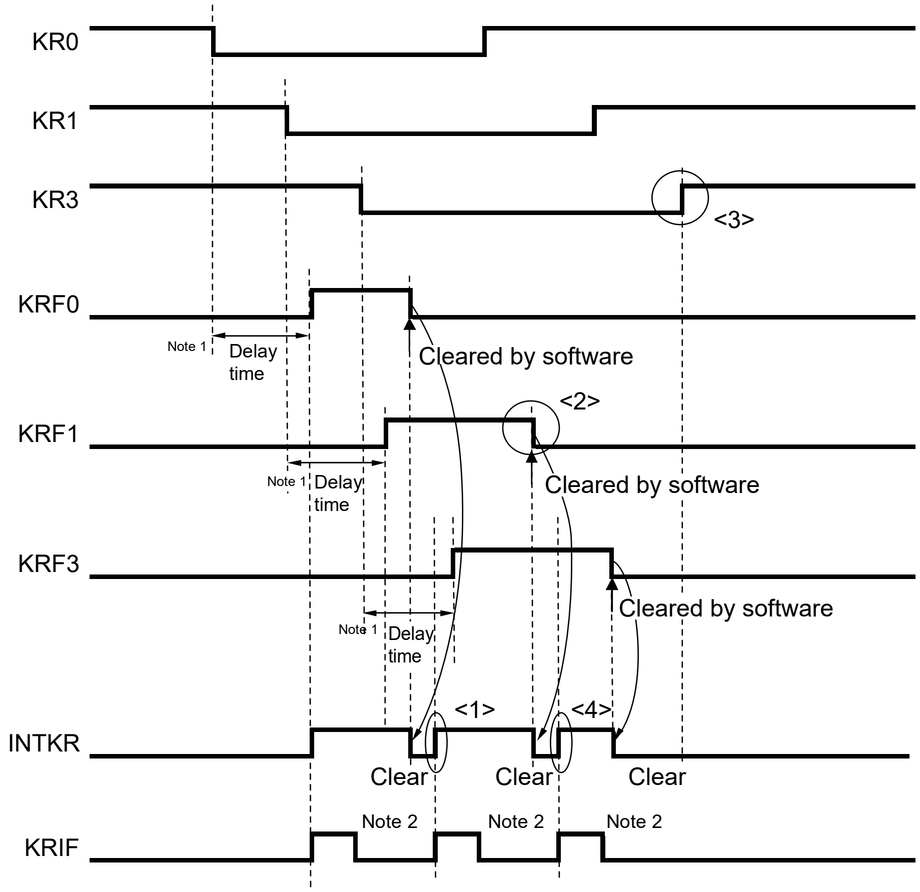


Notes 1. The maximum delay time is the maximum value of the high-level width and low-level width of the key interrupt input (see **27.4 AC Characteristics** for details).

2. Acknowledgment of vectored interrupt request or bit cleared by software

The operation when a valid edge is input to multiple key interrupt input pins is shown in Figure 15-9 below. A falling edge is also input to the KR1 and KR3 pins after a falling edge was input to the KR0 pin (when KREG = 0). The KRF1 bit is set when the KRF0 bit is cleared. A key interrupt (INTKR) is therefore generated one clock (f_{CLK}) after the KRF0 bit is cleared (<1> in the figure). Also, after a falling edge has been input to the KR3 pin, a low level continues to be input to this pin (<3> in the figure) until the KRF1 bit is cleared (<2> in the figure). A key interrupt (INTKR) is therefore generated one clock (f_{CLK}) after the KRF1 bit is cleared (<4> in the figure). It is thus possible to generate a key interrupt (INTKR) when a valid edge is input to multiple channels.

Figure 15-9. Operation of INTKR Signal When Key Interrupts Are Input to Multiple Channels (When KRMD = 1 and KREG = 0)



Notes 1. The maximum delay time is the maximum value of the high-level width and low-level width of the key interrupt input (see **27.4 AC Characteristics** for details).

2. Acknowledgment of vectored interrupt request or bit cleared by software

Remark f_{CLK}: CPU/peripheral hardware clock frequency

CHAPTER 16 STANDBY FUNCTION

16.1 Standby Function and Configuration

16.1.1 Standby function

The standby function reduces the operating current of the system, and the following three modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator, high-speed on-chip oscillator, or subsystem clock oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and high-speed on-chip oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

(3) SNOOZE mode

In the case of CSI00 or UART0 data reception and an A/D conversion request by the timer trigger signal (the interrupt request signal (INTRTC/INTIT)), the STOP mode is exited, the CSI00 or UART0 data is received without operating the CPU, and A/D conversion is performed. This can only be specified when the high-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (f_{CLK}).

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the CPU is operating on the main system clock. Do not set to the STOP mode while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction (except SNOOZE mode setting unit).

3. When using CSI00, UART0, or the A/D converter in the SNOOZE mode, set up serial standby control register 0 (SSC0) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 12.3 Registers Controlling Serial Array Unit and 11.3 Registers Used in A/D Converter.

4. The following sequence is recommended for power consumption reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

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 22 OPTION BYTE.

16.2 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 11 A/D CONVERTER** and **CHAPTER 12 SERIAL ARRAY UNIT**.

16.3 Standby Function Operation

16.3.1 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 16-1. Operating Statuses in HALT Mode (1/2)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on High-Speed On-Chip Oscillator Clock (f_{IH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Operation continues (cannot be stopped)	Operation disabled	
		f_X	Operation disabled	Operation continues (cannot be stopped)	Cannot operate
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained		
		f_{EXS}			
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) ● WUTMMCK0 = 1: Oscillates ● WUTMMCK0 = 0 and WDTON = 0: Stops ● WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates ● WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Code flash memory					
Data flash memory					
RAM					
Port (latch)			Status before HALT mode was set is retained		
Timer array unit			Operable		
Real-time clock (RTC)					
12-bit interval timer					
Watchdog timer			See **CHAPTER 10 WATCHDOG TIMER**		
Clock output/buzzer output			Operable		
A/D converter					
Serial array unit (SAU)					
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.)		
Power-on-reset function			Operable		
Voltage detection function					
External interrupt					
Key interrupt function					
CRC operation function	High-speed CRC				
	General-purpose CRC		Operation stopped		
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_{IH}: High-speed on-chip oscillator clock
f_X: X1 clock
f_{XT}: XT1 clock
f_{IL}: Low-speed on-chip oscillator clock
f_{EX}: External main system clock
f_{EXS}: External subsystem clock

Table 16-1. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting Item			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock	
			When CPU Is Operating on XT1 Clock (f_{XT})	When CPU Is Operating on External Subsystem Clock (f_{EXS})
System clock			Clock supply to the CPU is stopped	
	Main system clock	f_{IH}	Operation disabled	
		f_X		
		f_{EX}		
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)	Cannot operate
		f_{EXS}	Cannot operate	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) ● WUTMMCK0 = 1: Oscillates ● WUTMMCK0 = 0 and WDTON = 0: Stops ● WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates ● WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops	
CPU			Operation stopped	
Code flash memory				
Data flash memory				
RAM				
Port (latch)			Status before HALT mode was set is retained	
Timer array unit			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
Real-time clock (RTC)			Operable	
12-bit interval timer				
Watchdog timer			See **CHAPTER 10 WATCHDOG TIMER**	
Clock output/buzzer output			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
A/D converter			Operation disabled	
Serial array unit (SAU)			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
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.)	
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	
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_{IH}:	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_{EXS}:	External subsystem clock

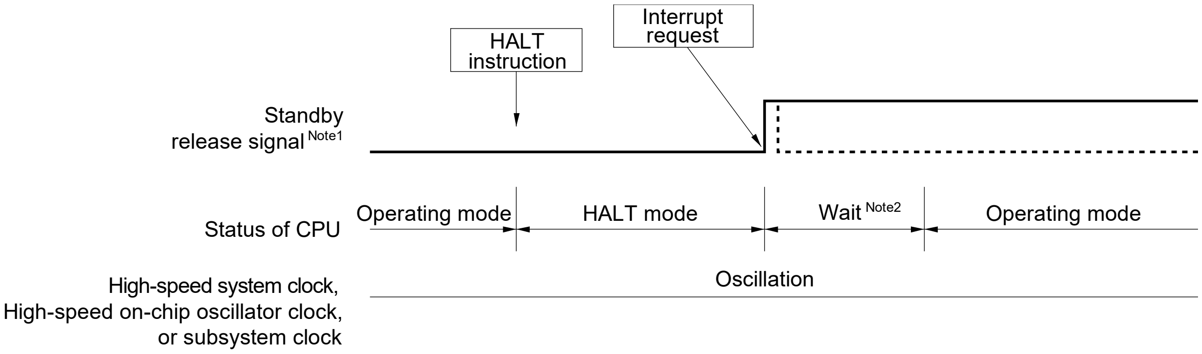
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 16-1. HALT Mode Release by Interrupt Request Generation



Notes 1. For details of the standby release signal, see Figure 14-1.

2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out

Main system clock:	15 to 16 clock
Subsystem clock (RTCLPC = 0):	10 to 11 clock
Subsystem clock (RTCLPC = 1):	11 to 12 clock

- When vectored interrupt servicing is not carried out

Main system clock:	9 to 10 clock
Subsystem clock (RTCLPC = 0):	4 to 5 clock
Subsystem clock (RTCLPC = 1):	5 to 6 clock

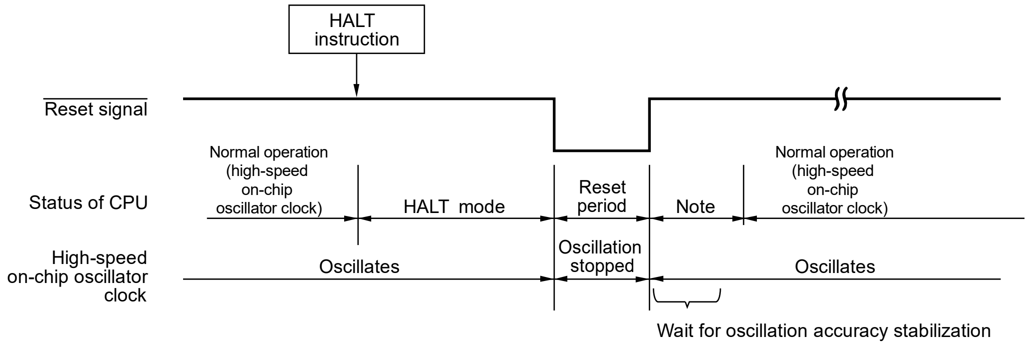
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

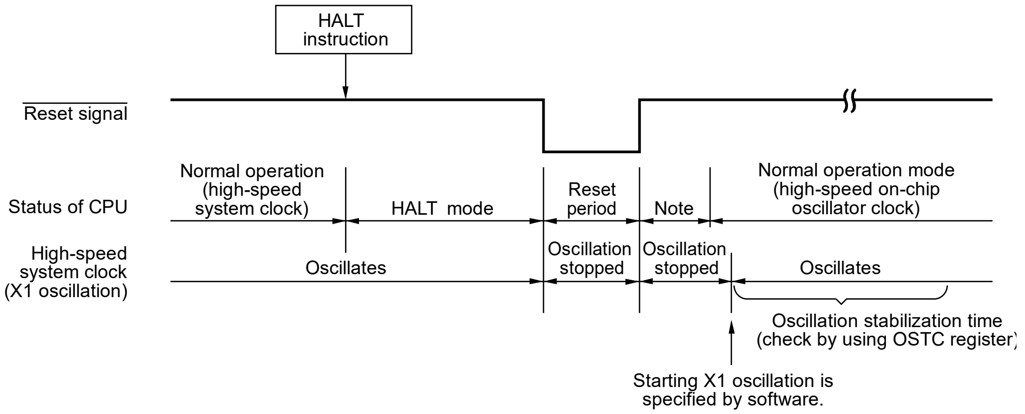
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 16-2. HALT Mode Release by Reset (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock



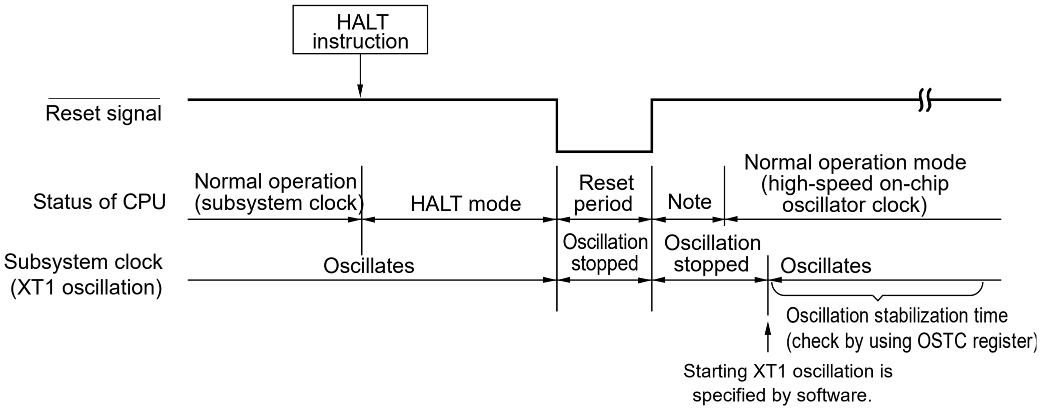
(2) When high-speed system clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 17 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 18 POWER-ON-RESET CIRCUIT**.

Figure 16-2. HALT Mode Release by Reset (2/2)

(3) When subsystem clock is used as CPU clock



Note For the reset processing time, see **CHAPTER17 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 18 POWER-ON-RESET CIRCUIT**.

16.3.2 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 main system clock.

Cautions 1. Because the interrupt request signal is used to clear the STOP mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the STOP mode is immediately cleared if set. Thus, when a STOP instruction is executed in this situation, the system returns to its normal operating mode as soon as the wait time set by using the oscillation stabilization time select register (OSTS) has elapsed. Note that the operating current during this period is the same as in the HALT mode because the clock is not stopped.

2. When using CSI00, UART0, or the A/D converter in the SNOOZE mode, set up serial standby control register 0 (SSC0) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 12.3 Registers Controlling Serial Array Unit and 11.3 Registers Used in A/D Converter.

The operating statuses in the STOP mode are shown below.

Table 16-2. Operating Statuses in STOP Mode

STOP Mode Setting / Item			When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on High-Speed On-Chip Oscillator Clock (f_{IH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Stopped		
		f_X			
		f_{EX}			
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained		
		f_{EXS}			
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) ● WUTMMCK0 = 1: Oscillates ● WUTMMCK0 = 0 and WDTON = 0: Stops ● WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates ● WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Code flash memory					
Data flash memory			Operation stopped		
RAM			Operation stopped		
Port (latch)			Status before STOP mode was set is retained		
Timer array unit			Operation disabled		
Real-time clock (RTC)			Operable		
12-bit interval timer					
Watchdog timer			See **CHAPTER 10 WATCHDOG TIMER**		
Clock output/buzzer output			Operates when the subsystem clock is selected as the clock source for counting and the RTCLPC bit is 0 (operation is disabled when a clock other than the subsystem clock is selected and the RTCLPC bit is not 0).		
A/D converter			Wakeup operation is enabled (switching to the SNOOZE mode)		
Serial array unit (SAU)			Wakeup operation is enabled only for CSI00 and UART0 (switching to the SNOOZE mode) Operation is disabled for anything other than CSI00 and UART0		
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.)		
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 Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.

f_{IH}:	High-speed on-chip oscillator clock	f_{IL}:	Low-speed on-chip oscillator clock
f_X:	X1 clock	f_{EX}:	External main system clock
f_{XT}:	XT1 clock	f_{EXS}:	External subsystem clock

Cautions 1. To stop the low-speed on-chip oscillator clock in the STOP mode, must previously be set an option byte to stop the watchdog timer operation in the HALT/STOP mode (bit 0 (WDSTBYON) of 000C0H = 0).

2. To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the high-speed on-chip oscillator clock before the execution of the STOP instruction. Before changing the CPU clock from the high-speed on-chip oscillator clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).

(2) STOP mode release

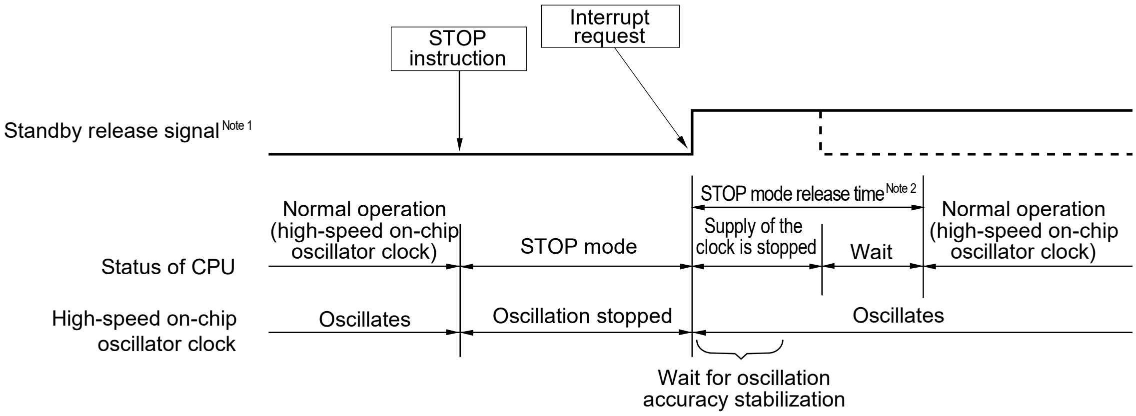
The STOP mode can be released by the following two sources.

(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 16-3. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock



Notes 1. For details of the standby release signal, see Figure 14-1.

2. STOP mode release time

Supply of the clock is stopped: 18 μs to 65 μs

Wait

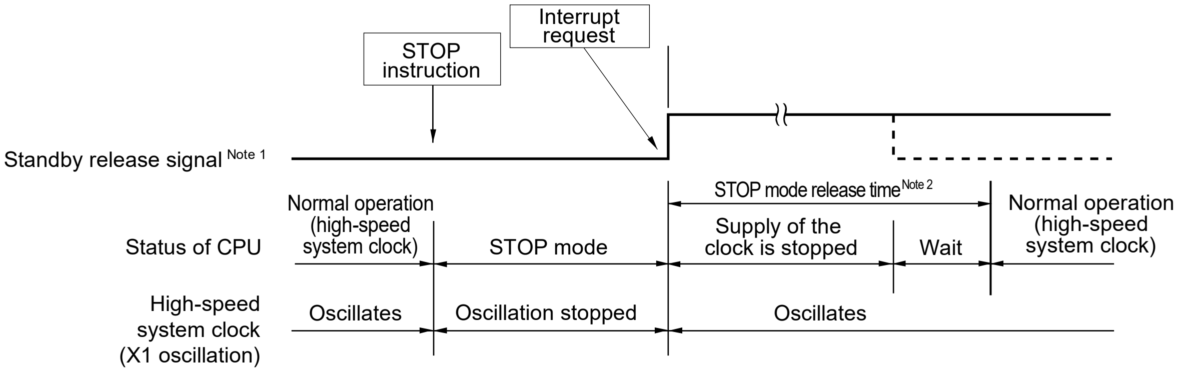
- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Remarks 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 16-3. STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock



Notes 1. For details of the standby release signal, see Figure 14-1.

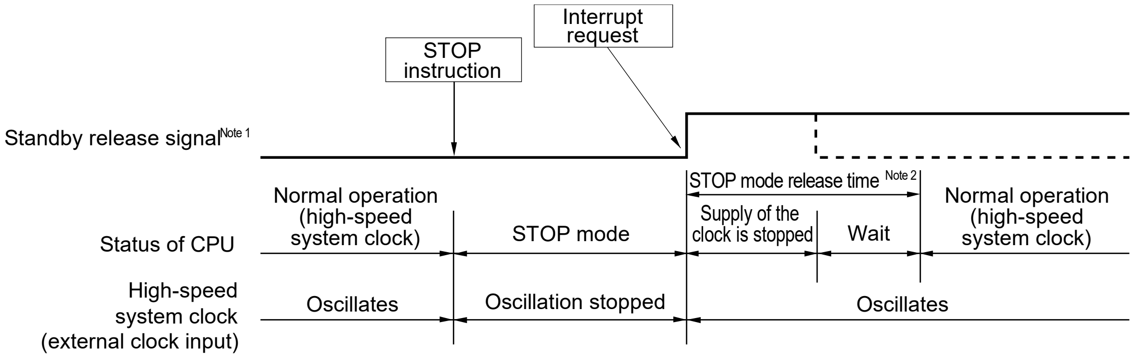
2. STOP mode release time

Supply of the clock is stopped: 18 μs to "whichever is longer 65 μs and the oscillation stabilization time (set by OSTS)"

Wait

- When vectored interrupt servicing is carried out: 10 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



Notes 1. For details of the standby release signal, see Figure 14-1.

2. STOP mode release time

Supply of the clock is stopped: 18 μs to 65 μs

Wait

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Caution **To reduce the oscillation stabilization time after release from the STOP mode while CPU operates based on the high-speed system clock (X1 oscillation), switch the clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.**

Remarks 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

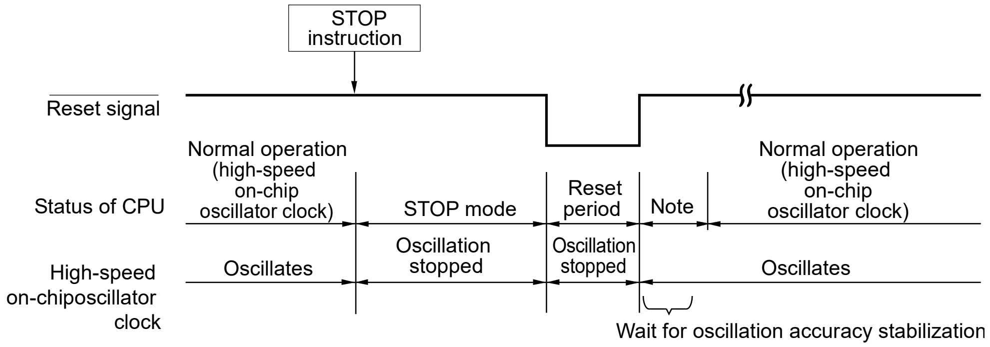
2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

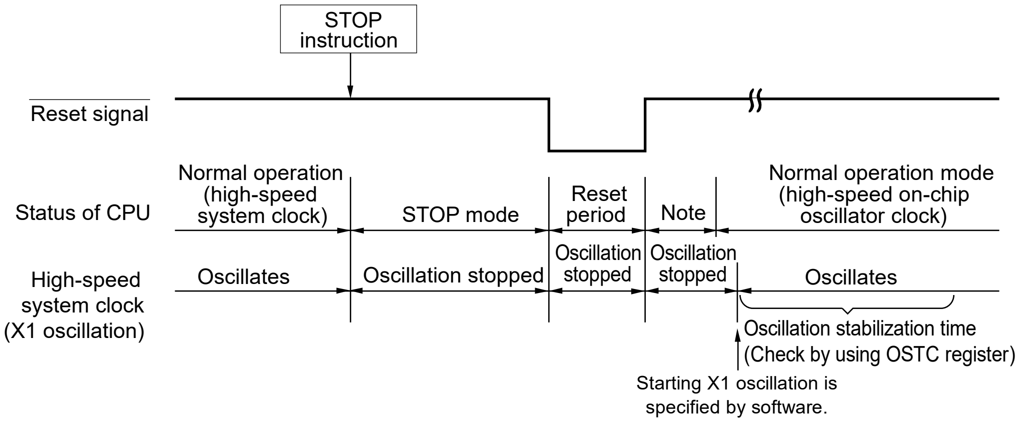
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 16-4. STOP Mode Release by Reset

(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 17 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 18 POWER-ON-RESET CIRCUIT**.

16.3.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can only be specified for CSI00, UART0, or the A/D converter. Note that this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock.
When using CSI00 or UART0 in the SNOOZE mode, set the SWCm bit of the serial standby control register m (SSCm) to 1 immediately before switching to the STOP mode. For details, see **12.3 Registers Controlling Serial Array Unit**.
When using the A/D converter in the SNOOZE mode, set the AWC bit of the A/D converter mode register 2 (ADM2) to 1 immediately before switching to the STOP mode. For details, see **11.3 Registers Used in A/D Converter**.

In SNOOZE mode transition, wait status to be only following time.

Transition time from STOP mode to SNOOZE mode: 18 μs to 65 μs

Remark Transition time from STOP mode to SNOOZE mode varies depending on the temperature conditions and the STOP mode period.

From SNOOZE to normal operation

- When vectored interrupt servicing is carried out:
  HS (High-speed main) mode : 4.99 to 9.44 μs + 7 clocks
  LS (Low-speed main) mode : 1.10 to 5.08 μs + 7 clocks
  LV (Low-voltage main) mode : 16.58 to 25.40 μs + 7 clocks
- When vectored interrupt servicing is not carried out:
  HS (High-speed main) mode : 4.99 to 9.44 μs + 1 clock
  LS (Low-speed main) mode : 1.10 to 5.08 μs + 1 clock
  LV (Low-voltage main) mode : 16.58 to 25.40 μs + 1 clock

The operating statuses in the SNOOZE mode are shown below.

Table 16-3. Operating Statuses in SNOOZE Mode

Item \ STOP Mode Setting			When Inputting CSI00/UART0 Data Reception Signal or A/D Converter Timer Trigger Signal While in STOP Mode
			When CPU Is Operating on High-Speed On-Chip Oscillator Clock (f_{IH})
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{IH}	Operation started
		f_X	Stopped
		f_{EX}	
	Subsystem clock	f_{XT}	Use of the status while in the STOP mode continues
		f_{EXS}	
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) ● WUTMMCK0 = 1: Oscillates ● WUTMMCK0 = 0 and WDTON = 0: Stops ● WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates ● WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops
CPU			Operation stopped
Code flash memory			
Data flash memory			
RAM			
Port (latch)			Use of the status while in the STOP mode continues
Timer array unit			Operation disabled
Real-time clock (RTC)			Operable
12-bit interval timer			
Watchdog timer			See **CHAPTER 10 WATCHDOG TIMER**
Clock output/buzzer output			Operates when the subsystem clock is selected as the clock source for counting and the RTCLPC bit is 0 (operation is disabled when a clock other than the subsystem clock is selected and the RTCLPC bit is not 0).
A/D converter			Operable
Serial array unit (SAU)			Operable only CSI00 and UART0 only. Operation disabled other than CSI00 and UART0.
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.)
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 Operation stopped: Operation is automatically stopped before switching to the SNOOZE mode.

Operation disabled: Operation is stopped before switching to the SNOOZE mode.

f_{IH}:	High-speed on-chip oscillator clock	f_{IL}:	Low-speed on-chip oscillator clock
f_X:	X1 clock	f_{EX}:	External main system clock
f_{XT}:	XT1 clock	f_{EXS}:	External subsystem clock

CHAPTER 17 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 stored 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 17-1.

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

Cautions 1. For an external reset, input a low level for 10 µs or more to the $\overline{\text{RESET}}$ pin.
To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for 10 us or more within the operating voltage range shown in 27.4 AC Characteristics, and then input a high level to the pin.

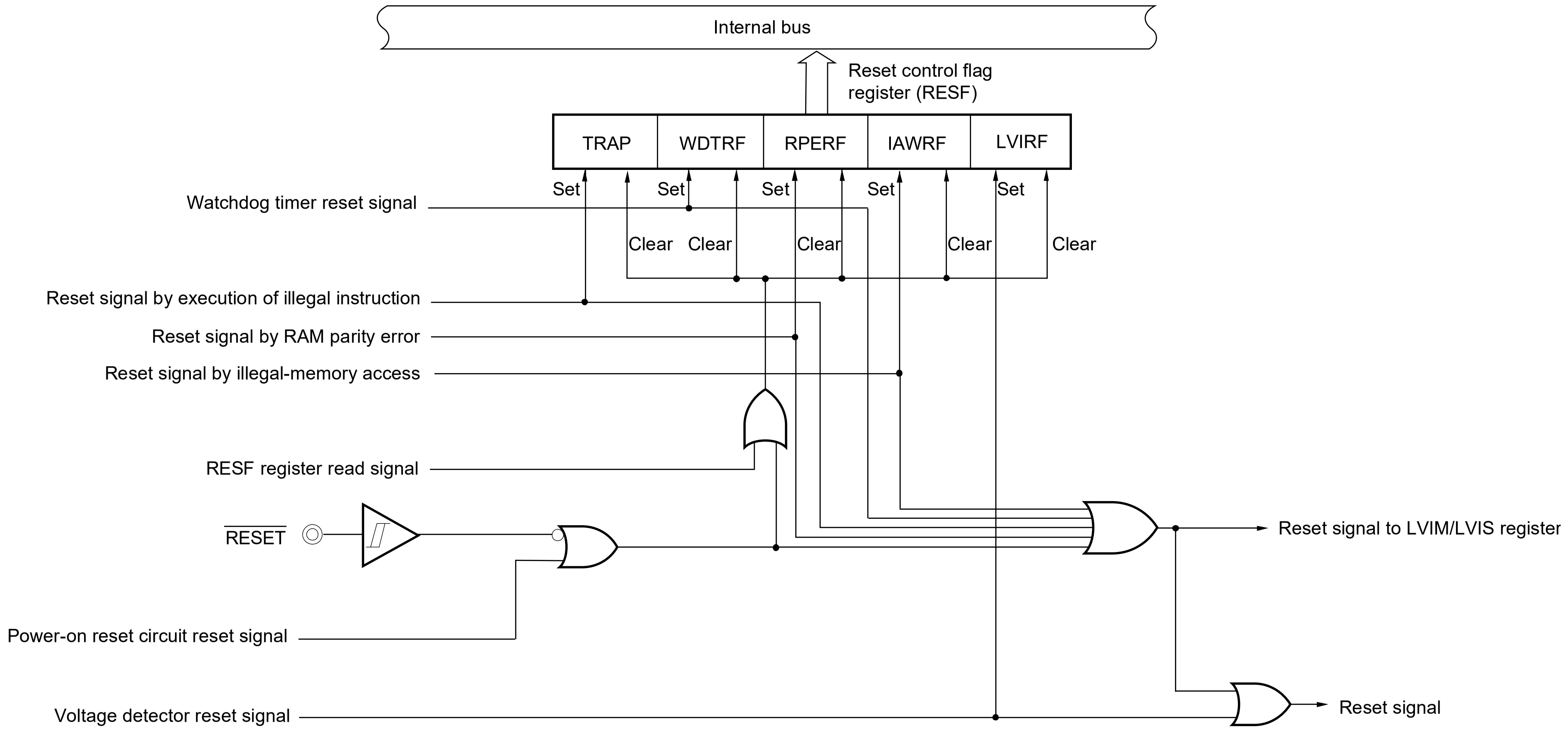
2. During reset input, the X1 clock, XT1 clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating. External main system clock input and external subsystem clock input become invalid.

3. The port pins become the following state because each SFR and 2nd SFR are initialized after reset.

- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).**
- **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 17-1. Block Diagram of Reset Function



Caution **An LVD circuit internal reset does not reset the LVD circuit.**

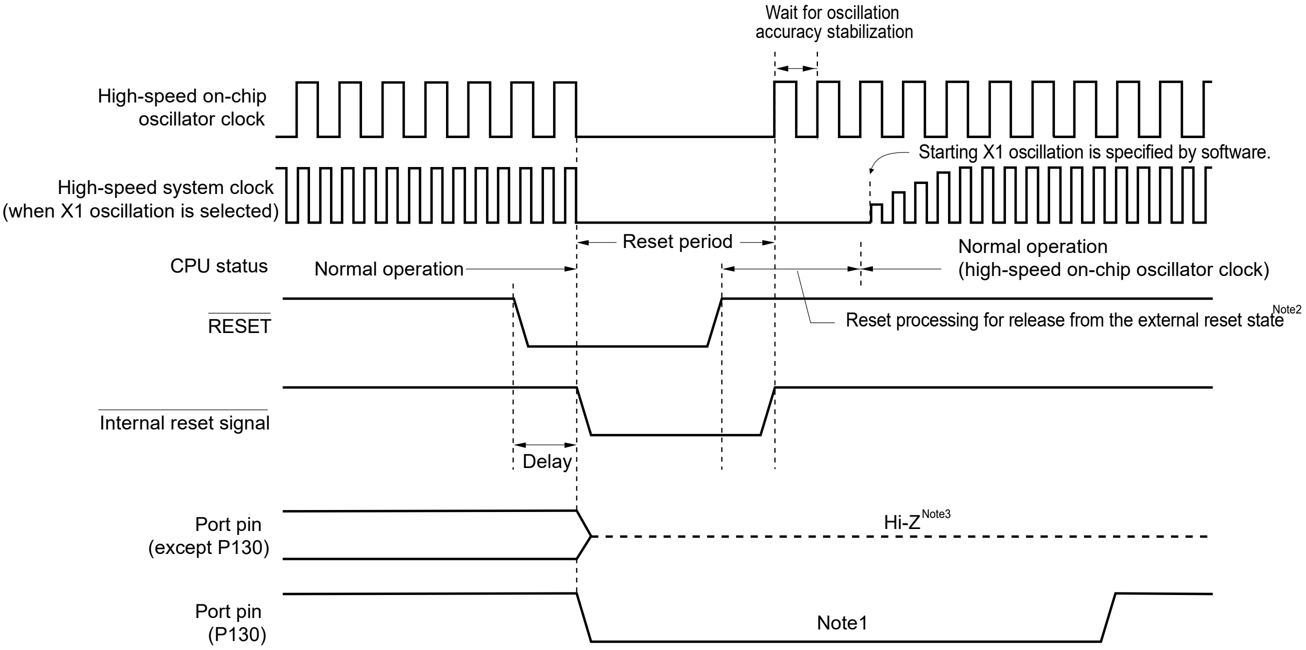
Remarks **1.** LVIM: Voltage detection register

2. LVIS: Voltage detection level register

17.1 Timing of Reset Operation

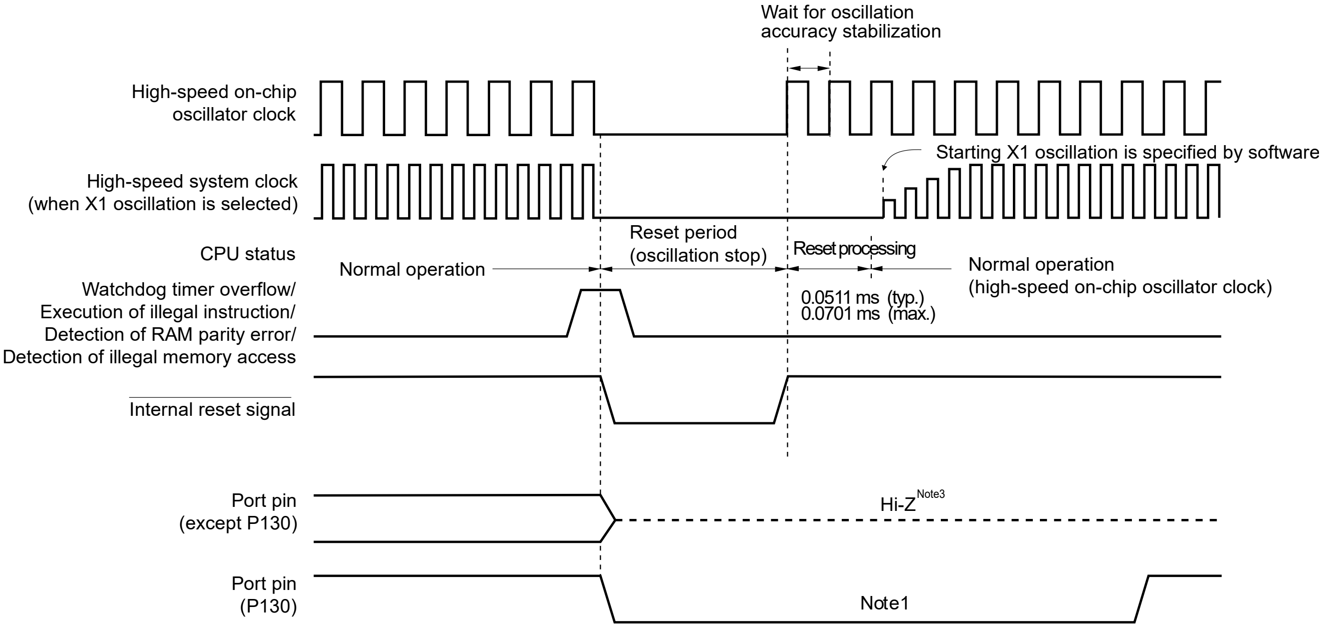
This LSI is reset by input of the low level on the $\overline{\text{RESET}}$ pin and released from the reset state by input of the high level on the $\overline{\text{RESET}}$ pin. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 17-2. Timing of Reset by $\overline{\text{RESET}}$ Input



Release from the reset state is automatic in the case of a reset due to a watchdog timer overflow, execution of an illegal instruction, detection of a RAM parity error, or detection of illegal memory access. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 17-3. Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal Memory Access



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

Notes 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.

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.

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 resistor).

Remark For the reset timing of the power-on-reset circuit and voltage detector, see **CHAPTER 18 POWER-ON-RESET CIRCUIT** and **CHAPTER 19 VOLTAGE DETECTOR**.

17.2 States of Operation During Reset Periods

Table 17-1 shows the states of operation during reset periods. Table 17-2 shows the states of the hardware after receiving a reset signal.

Table 17-1. States of Operation During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_X	Operation stopped (the X1 and X2 pins are input port mode)
		f_{EX}	Clock input invalid (the pin is input port mode)
	Subsystem clock	f_{XT}	Operation stopped (the XT1 and XT2 pins are input port mode)
		f_{EXS}	Clock input invalid (the pin is input port mode)
	f_{IL}		Operation stopped
CPU			Operation stopped
Code flash memory			Operation stopped
Data flash memory			Operation stopped
RAM			Operation stopped
Port (latch)			High impedance [Note]
Timer array unit			Operation stopped
Real-time clock (RTC)			
12-bit interval timer			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Serial array unit (SAU)			
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)
Power-on-reset function			Detection operation possible
Voltage detection function			Operation stopped
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** and **Remark** are listed on the next page.)

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 internal pull-up resistor).
- P130: Low level during the reset period

Remark

f_{IH}:	High-speed on-chip oscillator clock
f_X:	X1 oscillation clock
f_{EX}:	External main system clock
f_{XT}:	XT1 oscillation clock
f_{EXS}:	External subsystem clock
f_{IL}:	Low-speed on-chip oscillator clock

Table 17-2. 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 **3.2.4 Special function registers (SFRs)** and **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

17.3 Register for Confirming Reset Source

17.3.1 Reset Control Flag Register (RESF)

Many internal reset generation sources exist in the R7F0C001G/L, R7F0C002G/L. 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 17-4. 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 by illegal-memory access
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by voltage detector (LVD)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

Notes 1. The value after reset varies depending on the reset source. See Table 17-3.

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

Cautions 1. Do not read data by a 1-bit memory manipulation instruction.

2. When enabling RAM parity error resets (RPERDIS = 0), be sure to initialize the used RAM area at data access or the used RAM area + 10 bytes at execution of instruction from the RAM area.
Reset generation enables RAM parity error resets (RPERDIS = 0). For details, see 20.3.3 RAM parity error detection function.

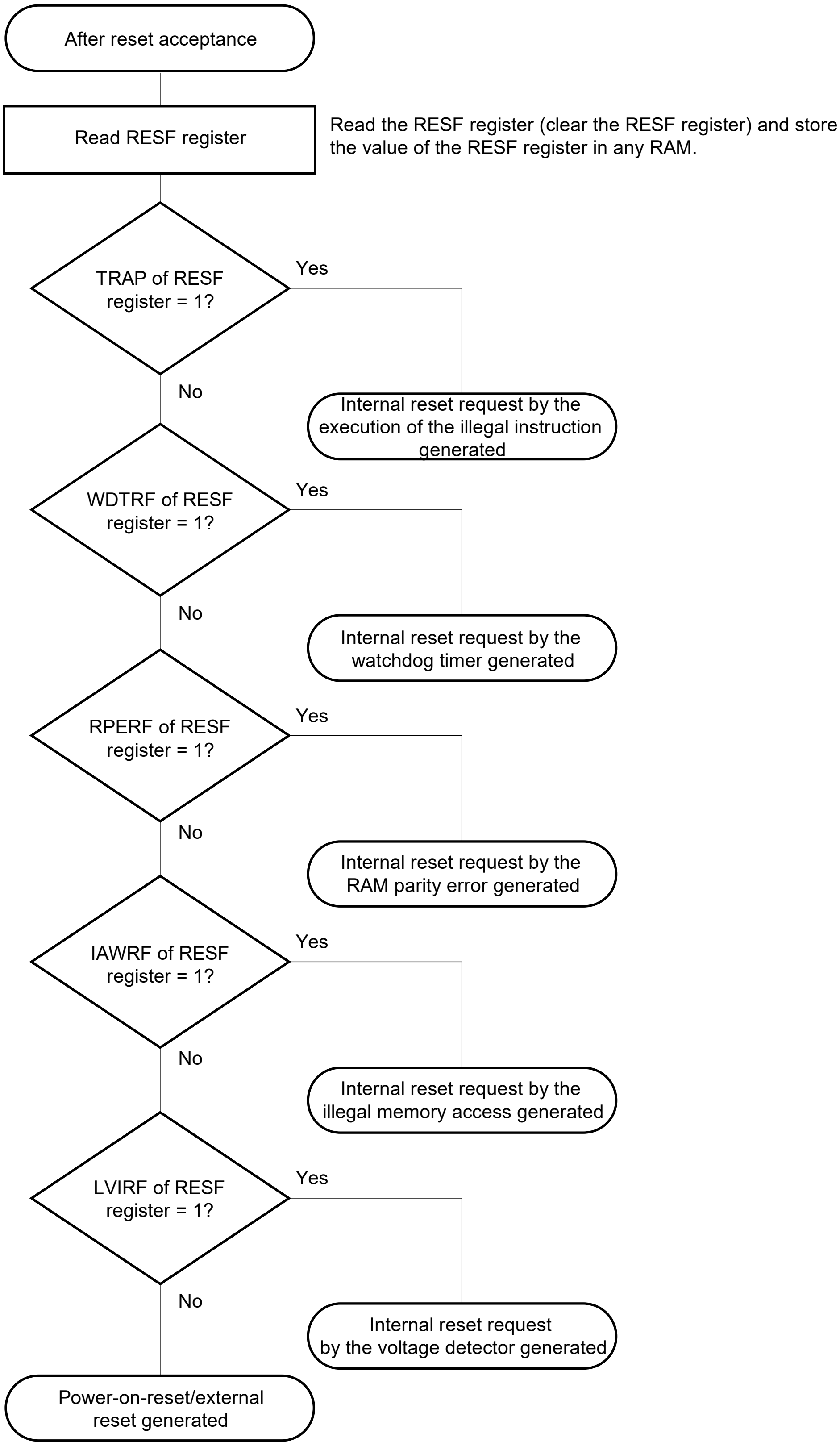
The status of the RESF register when a reset request is generated is shown in Table 17-3.

Table 17-3. 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 17-5 shows the procedure for checking a reset source.

Figure 17-5. Example of Procedure for Checking Reset Source



CHAPTER 18 POWER-ON-RESET CIRCUIT

18.1 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 **27.4 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 **27.4 or AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Caution If an internal reset signal is generated in the power-on-reset circuit, the reset control flag register (RESF) is cleared.

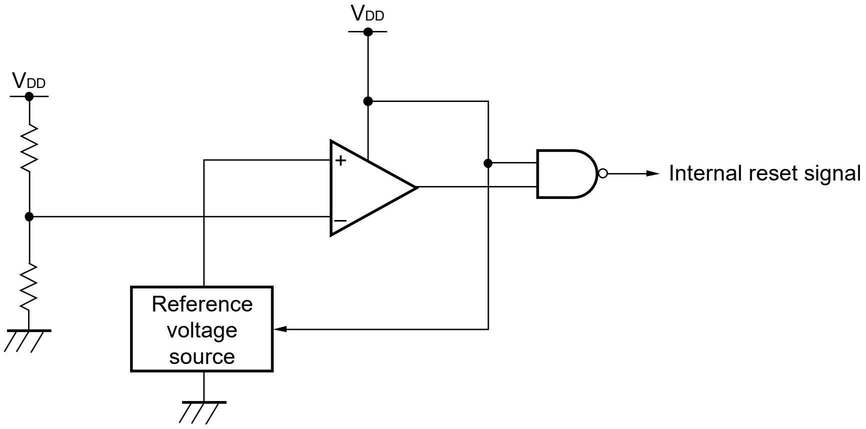
Remarks 1. The R7F0C001G/L, R7F0C002G/L incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (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 17 RESET FUNCTION**.

2. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **27.6.3 POR circuit characteristics**.

18.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 18-1.

Figure 18-1. Block Diagram of Power-on-reset Circuit

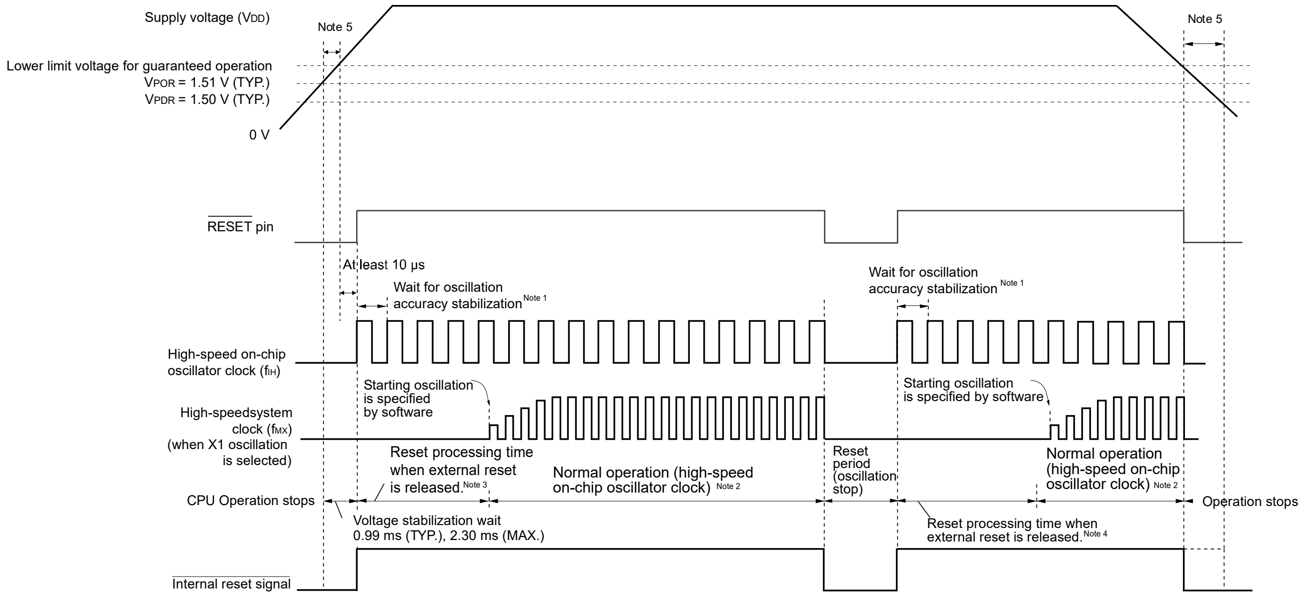


18.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown below.

Figure 18-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3)

(1) When the externally input reset signal on the $\overline{\text{RESET}}$ pin is used



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

2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

3. The time until normal operation starts includes the following reset processing time when the external reset is released (release from the first external reset 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.

With the LVD circuit in use: 0.672 ms (typ.), 0.832 ms (max.)
With the LVD circuit not in use: 0.399 ms (typ.), 0.519 ms (max.)

4. The reset processing times in the case of the second or subsequent external reset following release from the POR state are listed below.

With the LVD circuit in use: 0.531 ms (typ.), 0.675 ms (max.)
With the LVD circuit not in use: 0.259 ms (typ.), 0.362 ms (max.)

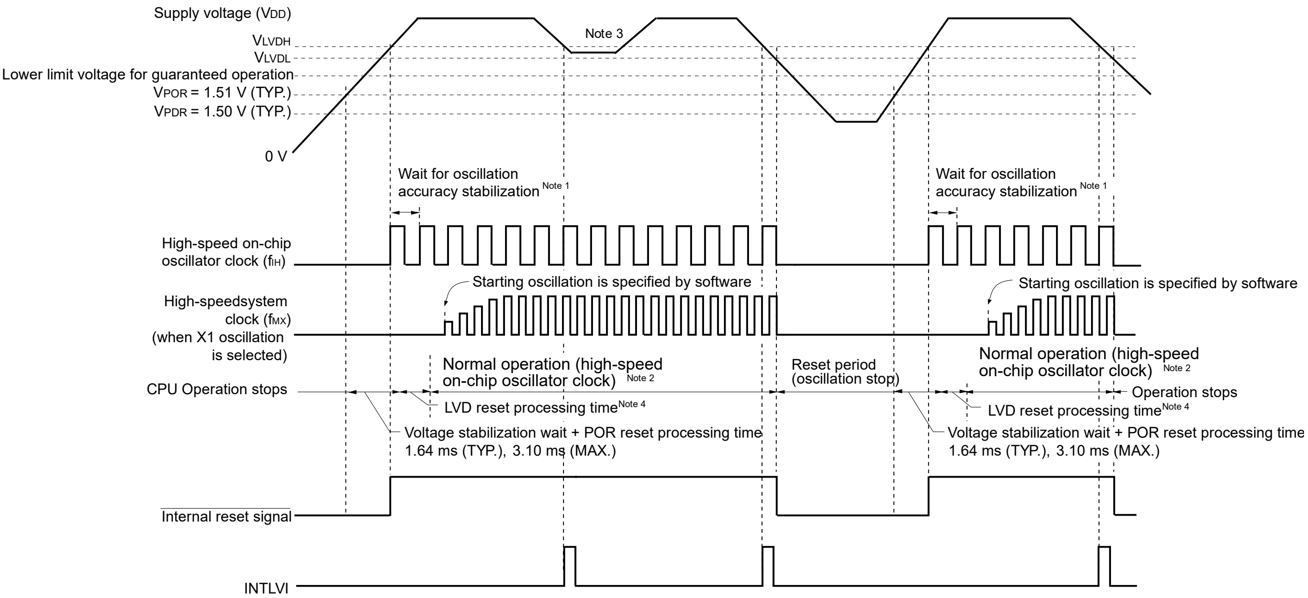
5. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **27.4 AC Characteristics**. This is done by controlling the externally input reset signal. After power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the voltage falls below the operating range. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Caution For power-on reset, be sure to use the externally input reset signal on the $\overline{\text{RESET}}$ pin when the LVD is off. For details, see CHAPTER 19 VOLTAGE DETECTOR.

Figure 18-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)

(2) LVD interrupt & reset mode (option byte 000C1: LVIMDS1, LVIMDS0 = 1, 0)



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

2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

3. After the first interrupt request signal (INTLVI) is generated, the LVIL and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, the initial settings should be made by software after the required save processing if the supply voltage might return to the high voltage detection level (V_{LVDH}) or higher without falling below the low voltage detection level (V_{LVDL}).

4. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVDH}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, typ.) is reached.

LVD reset processing time: 0 ms to 0.0701 ms (max.)

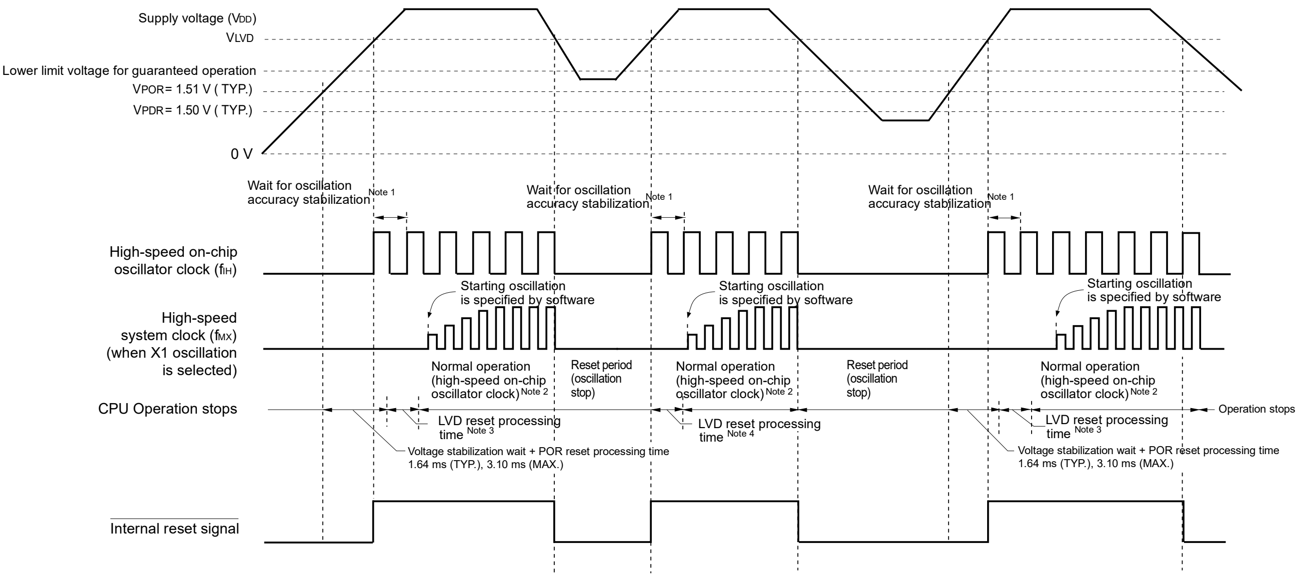
Remark V_{LVDH}, V_{LVDL}: LVD detection voltage

V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

Figure 18-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (3/3)

(3) LVD reset mode (option byte 000C1H: LVIMDS1 = 1, LVIMDS0 = 1)



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

2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

3. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{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.)

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 V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

CHAPTER 19 VOLTAGE DETECTOR

19.1 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}), and generates an internal reset or internal interrupt signal.
- The detection level for the power supply detection voltage (V_{LVDH}, V_{LVDL}) can be selected by using the option byte as one of 14 levels (For details, see **CHAPTER 22 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 **27.4 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).

(a) Interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0)

The two detection voltages (V_{LVDH}, V_{LVDL}) are selected by the option byte 000C1H. The high-voltage detection level (V_{LVDH}) is used for releasing resets and generating interrupts. The low-voltage detection level (V_{LVDL}) is used for generating resets.

(b) Reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1)

The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for generating/releasing resets.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)

The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for releasing resets/generating interrupts.

The reset and internal interrupt signals are generated in each mode as follows.

Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage falls, and an internal reset by detecting $V_{DD} < V_{LVDL}$. Releases an internal reset by detecting $V_{DD} \geq V_{LVDH}$.	Releases an internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an internal reset by detecting $V_{DD} < V_{LVD}$.	Retains the state of an internal reset by the LVD immediately after a reset until $V_{DD} \geq V_{LVD}$. Releases the LVD internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an interrupt request signal (INTLVI) by detecting $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ after the LVD internal reset is released.

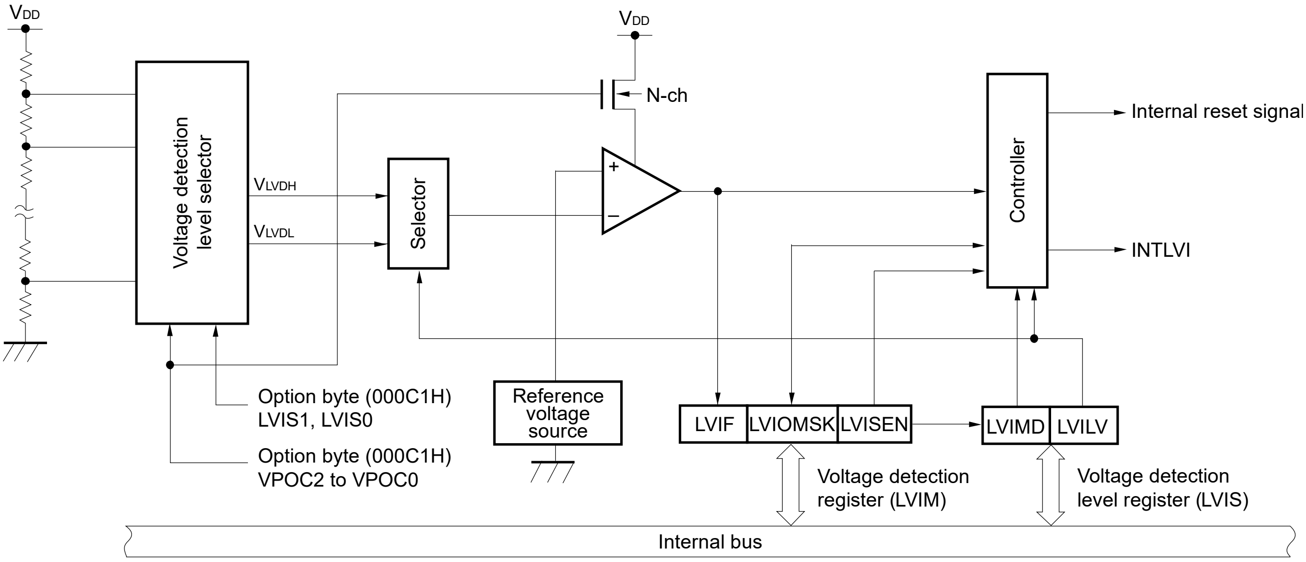
While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).

Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 17 RESET FUNCTION**.

19.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 19-1.

Figure 19-1. Block Diagram of Voltage Detector



19.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

19.3.1 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 19-2. 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})

Notes

1. The reset value changes depending on the reset source.
   If the LVIS register is reset by LVD, it is not reset but holds the current value. In other reset, LVISEN is cleared to 0.
2. Bits 0 and 1 are read-only.
3. LVISEN and LVIOMSK can only be set in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not change the initial value in other modes.
4. LVIOMSK bit is only automatically set to “1” during the following period 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

19.3.2 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 [Note1].

Figure 19-3. Format of Voltage Detection Level Select Register (LVIS)

Address: FFFAAH After reset: 00H/01H/81H [Note 1] R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD	0	0	0	0	0	0	LVILV

LVIMD [Note 2]	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVILV [Note 2]	LVD detection level
0	High-voltage detection level (V_{LVDH})
1	Low-voltage detection level (V_{LVDL} or V_{LVDL})

Notes 1. The reset value changes depending on the reset source and the setting of the option byte.
This register is not cleared (00H) by LVD reset.
The generation of reset signal other than an LVD reset sets as follows.
- When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H
- When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H
- When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H

2. Writing “0” can only be allowed in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not set LVIMD and LVILV in other cases. The value is switched automatically when reset or interrupt is generated in the interrupt & reset mode.

Cautions 1. Rewrite the value of the LVIS register according to Figures 19-7 and 19-8.

2. Specify the LVD operation mode and detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) of each mode by using the option byte 000C1H. Figure 19-4 shows the format of the user option byte (000C1H). For details about the option byte, see CHAPTER 22 OPTION BYTE.

Figure 19-4. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H) (1/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt & reset mode)

Detection voltage			Option byte setting value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
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		
3.75 V	3.67 V					0	0		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
–			Setting of values other than above is prohibited.						

- LVD setting (reset mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
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		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited.						

Caution Be sure to set bit 4 to 1.

Figure 19-4. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H) (2/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
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		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited.						

- LVD off (use of external reset input via $\overline{\text{RESET}}$ pin)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
–	–	1	×	×	×	×	0/1	1
–		Setting of values other than above is prohibited.						

Cautions 1. Be sure to set bit 4 to 1.

2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 27.4 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).

Remarks 1. ×: don’t care

2. For details on the LVD circuit, see **CHAPTER 19 VOLTAGE DETECTOR**.

3. The detection voltage is a TYP. value. For details, see **27.6.4 LVD circuit characteristics**.

19.4 Operation of Voltage Detector

19.4.1 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.

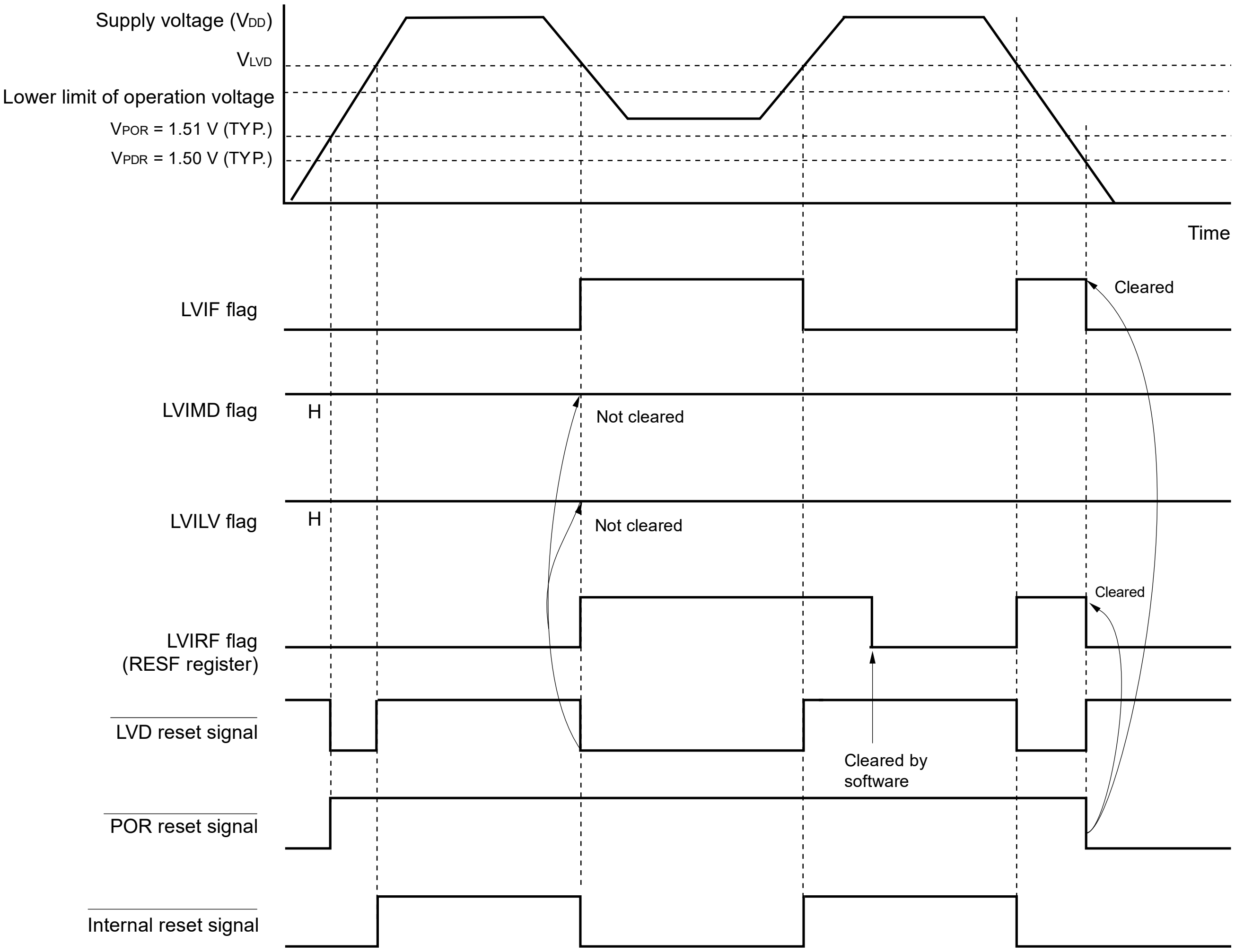
The operation is started in the following initial setting state when the reset mode is set.

- 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 19-5 shows the timing of the internal reset signal generated in the LVD reset mode.

Figure 19-5. Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)



Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

19.4.2 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}).

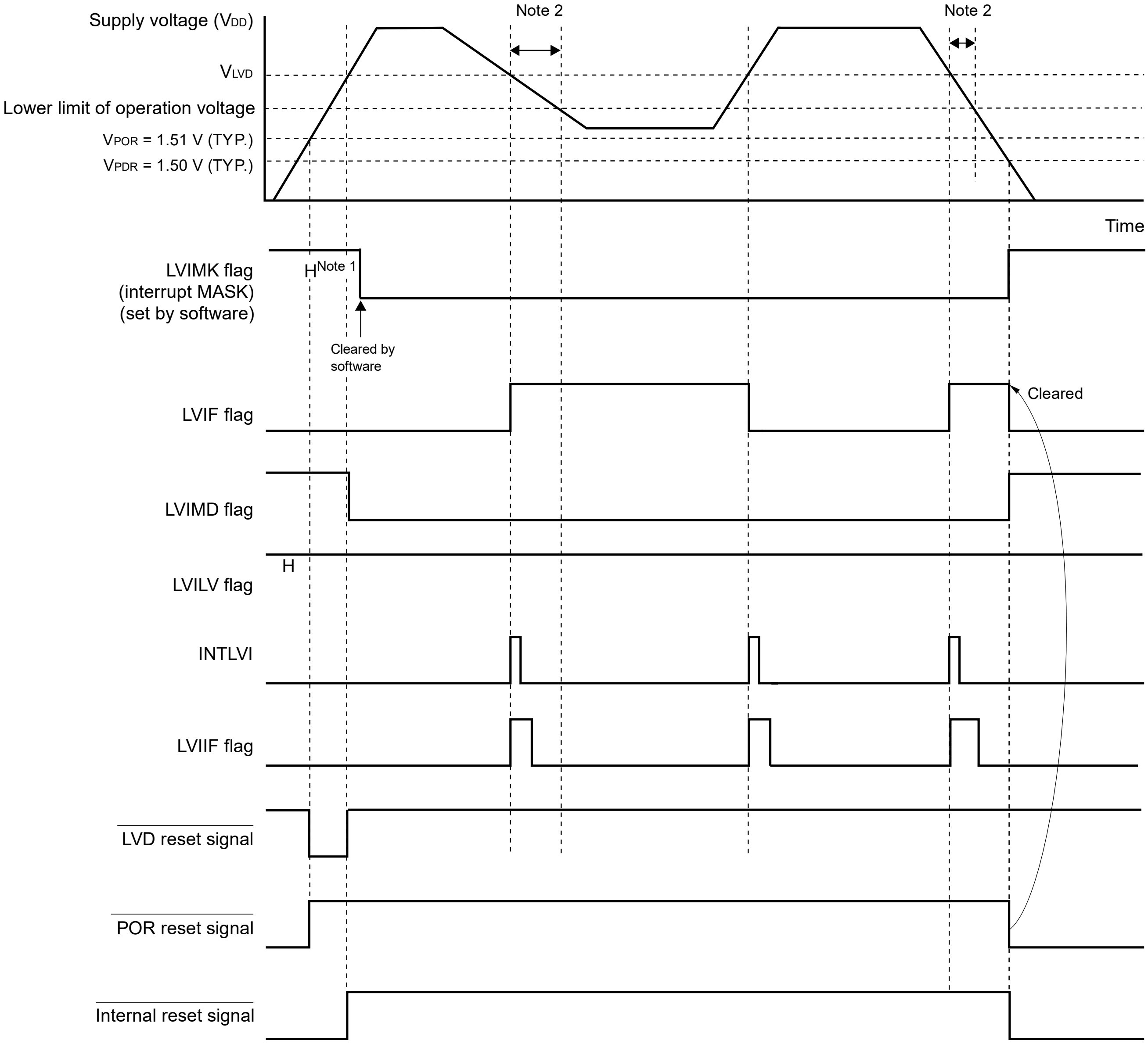
- Operation in LVD interrupt mode

  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 **27.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Figure 19-6 shows the timing of the interrupt request signal generated in the LVD interrupt mode.

Figure 19-6. Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **27.4 AC characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

19.4.3 When used as interrupt and reset mode

- When starting operation

  Specify the operation mode (the interrupt and 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.

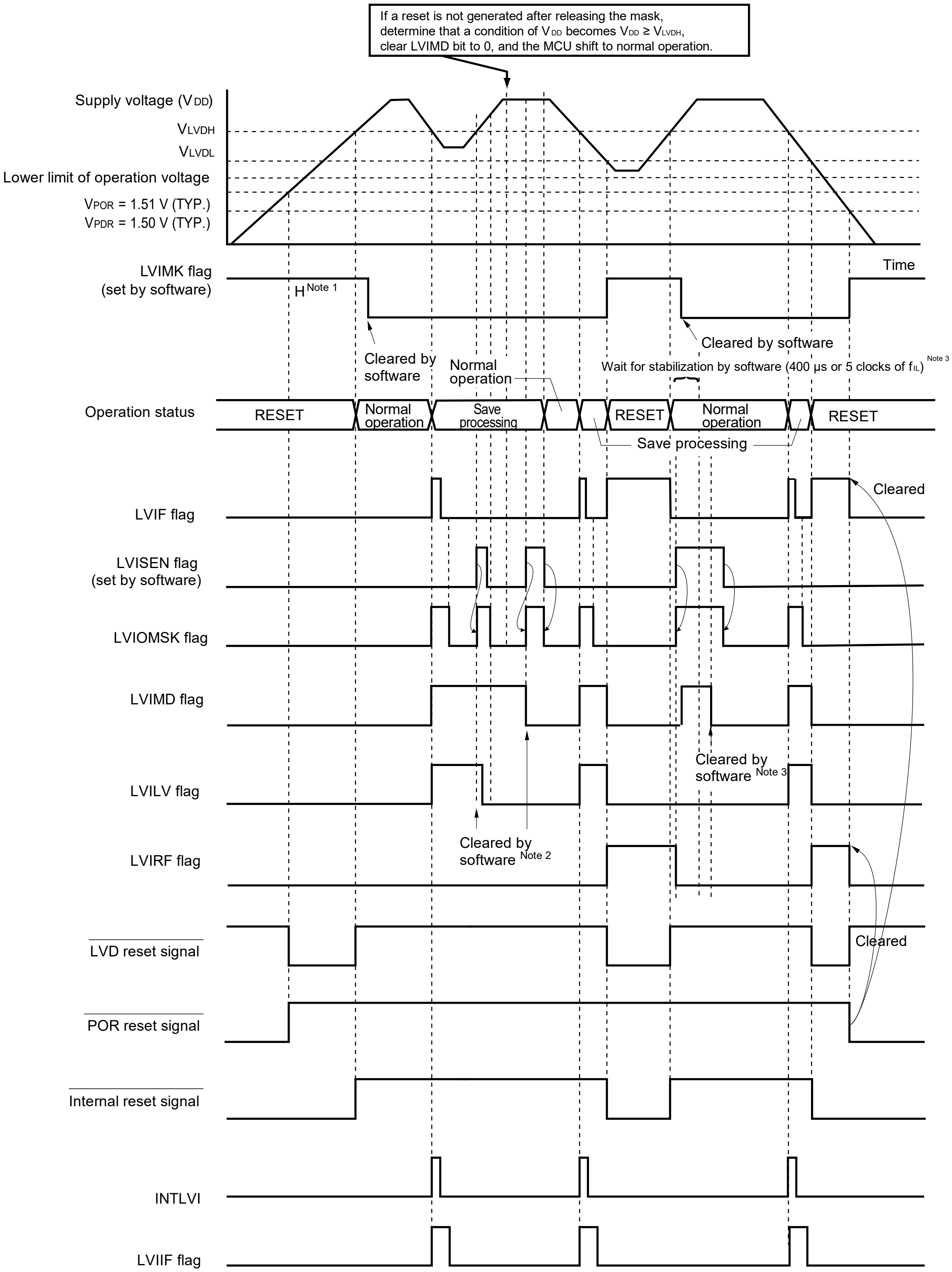
  - Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
  - The initial value of the voltage detection level select register (LVIS) is set to 00H.
    Bit 7 (LVIMD) is 0 (interrupt mode).
    Bit 0 (LVILV) is 0 (high-voltage 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 19-8 Processing Procedure After an Interrupt Is Generated** and **Figure 19-9 Initial Setting of Interrupt and Reset Mode**.

Figure 19-7 shows the timing of the internal reset signal and interrupt signal generated in the LVD interrupt & reset mode.

Figure 19-7. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



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

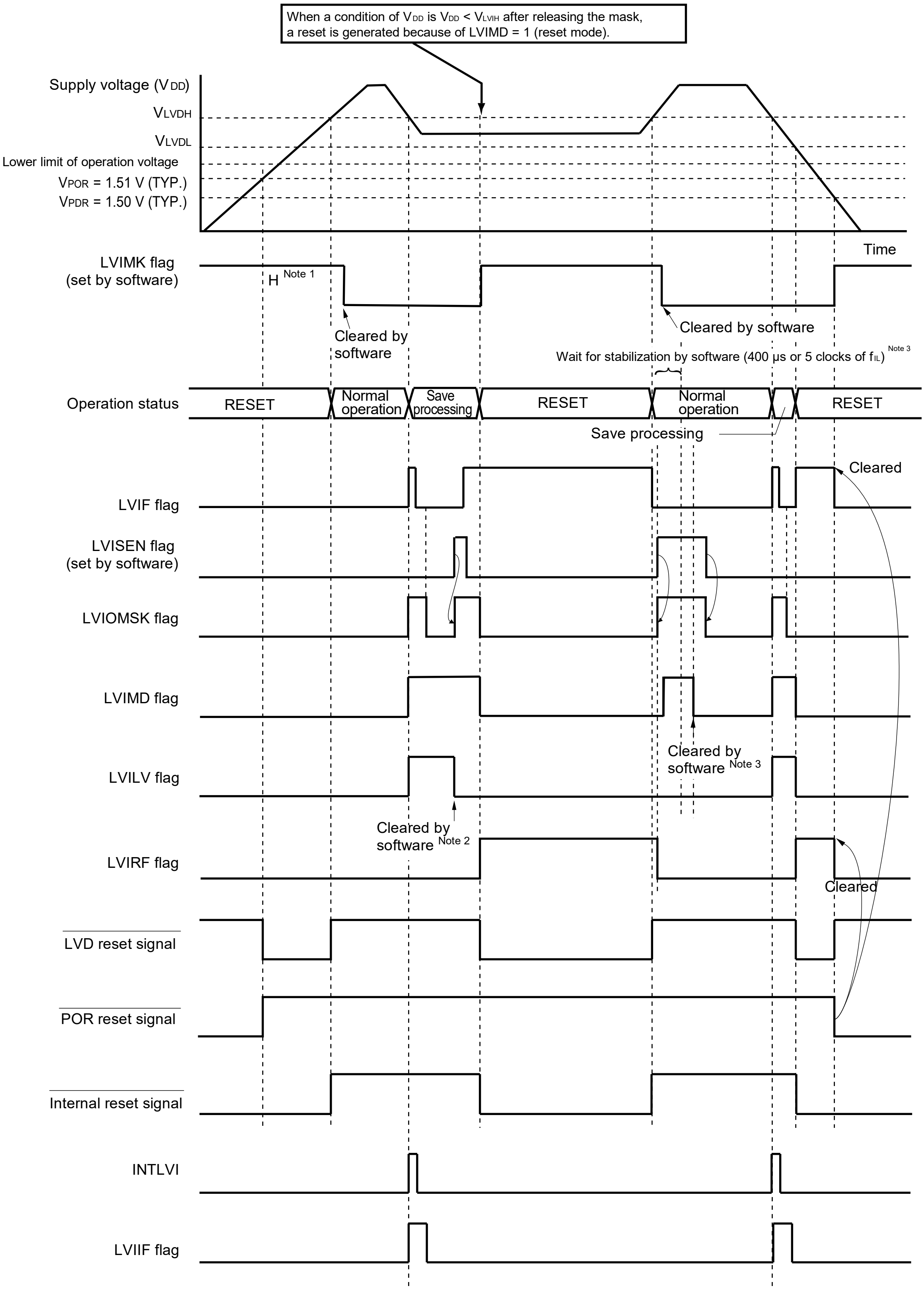
Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. After an interrupt is generated, perform the processing according to **Figure 19-8 Processing Procedure After an Interrupt Is Generated**.

3. After a reset is released, perform the processing according to **Figure 19-9 Initial Setting of Interrupt and Reset Mode**.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 19-7. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)



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

Notes 1. The LVIMK flag is set to “1” by reset signal generation.

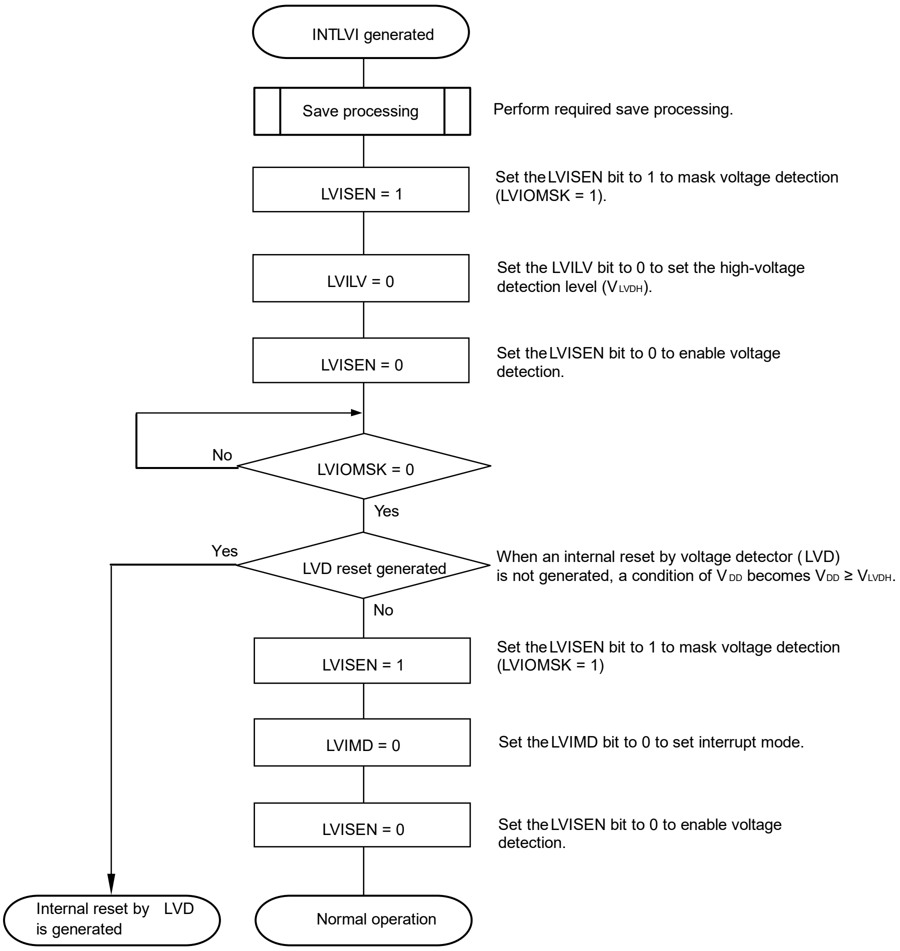
2. After an interrupt is generated, perform the processing according to **Figure 19-8 Processing Procedure After an Interrupt Is Generated**.

3. After a reset is released, perform the processing according to **Figure 19-9 Initial Setting of Interrupt and Reset Mode**.

Remark V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

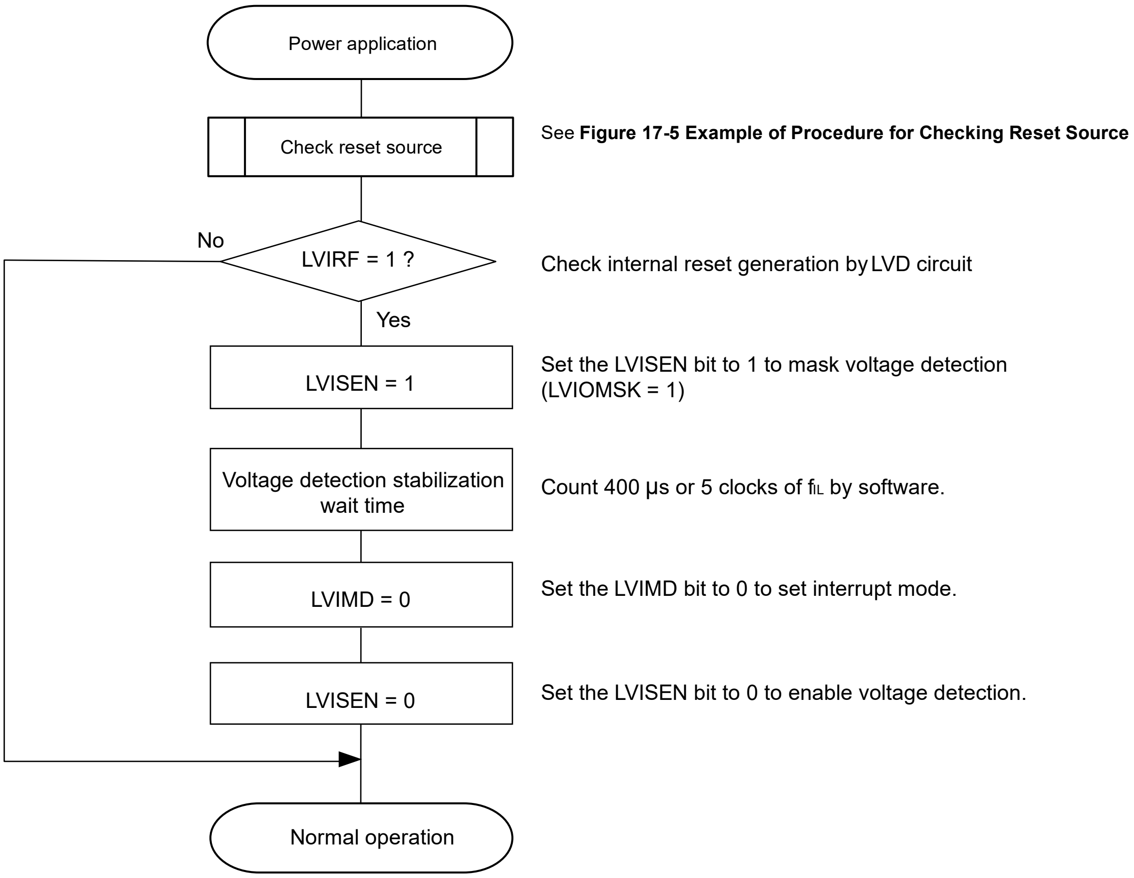
Figure 19-8. Processing Procedure After an Interrupt Is Generated



When setting an interrupt and reset mode (LVIMDS1, LVIMDS0 = 1, 0), voltage detection stabilization wait time for 400 μs or 5 clocks of f_{IL} is necessary after LVD reset is released (LVIRF = 1). After waiting until voltage detection stabilizes, (0) clear the LVIMD bit for initialization. While voltage detection stabilization wait time is being counted and when the LVIMD bit is rewritten, set LVISEN to 1 to mask a reset or interrupt generation by LVD.

Figure 19-9 shows the procedure for initial setting of interrupt and reset mode.

Figure 19-9. Initial Setting of Interrupt and Reset Mode



Remark f_{IL}: Low-speed on-chip oscillator clock frequency

19.5 Cautions for Voltage Detector

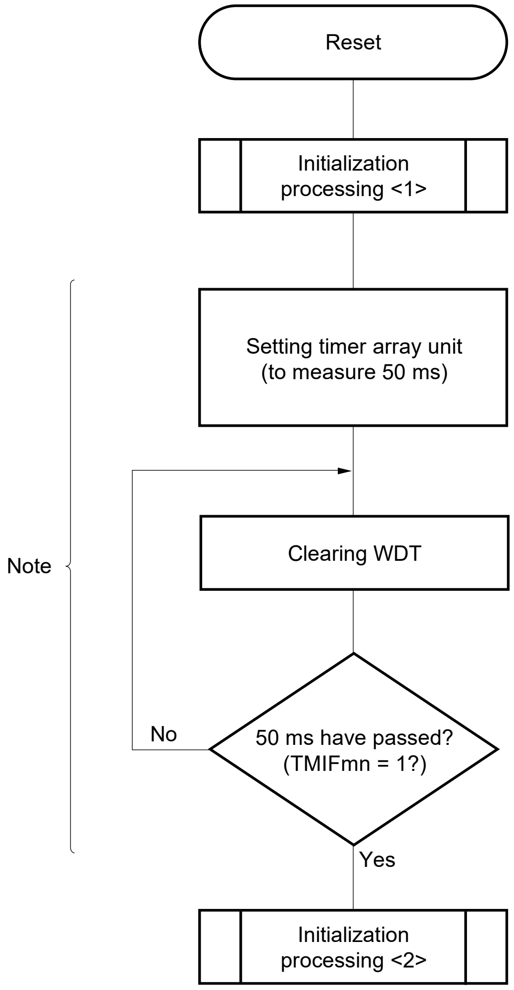
(1) Voltage fluctuation when power is supplied

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 19-10. Example of Software Processing If Supply Voltage Fluctuation is 50 ms or Less in Vicinity of LVD Detection Voltage



See Figure 17-5 Example of Procedure for Checking Reset Source.

; e.g. f_{CLK} = High-speed on-chip oscillator clock (4.04 MHz (MAX.))
Source: f_{MCK} = (4.04 MHz (MAX.))/2^8,
where comparison value = 789: ≈ 50 ms
Timer starts (TSmn = 1).

; Initial setting for port.
Setting of division ratio of system clock,
such as setting of timer or A/D converter.

Note If reset is generated again during this period, initialization processing <2> is not started.

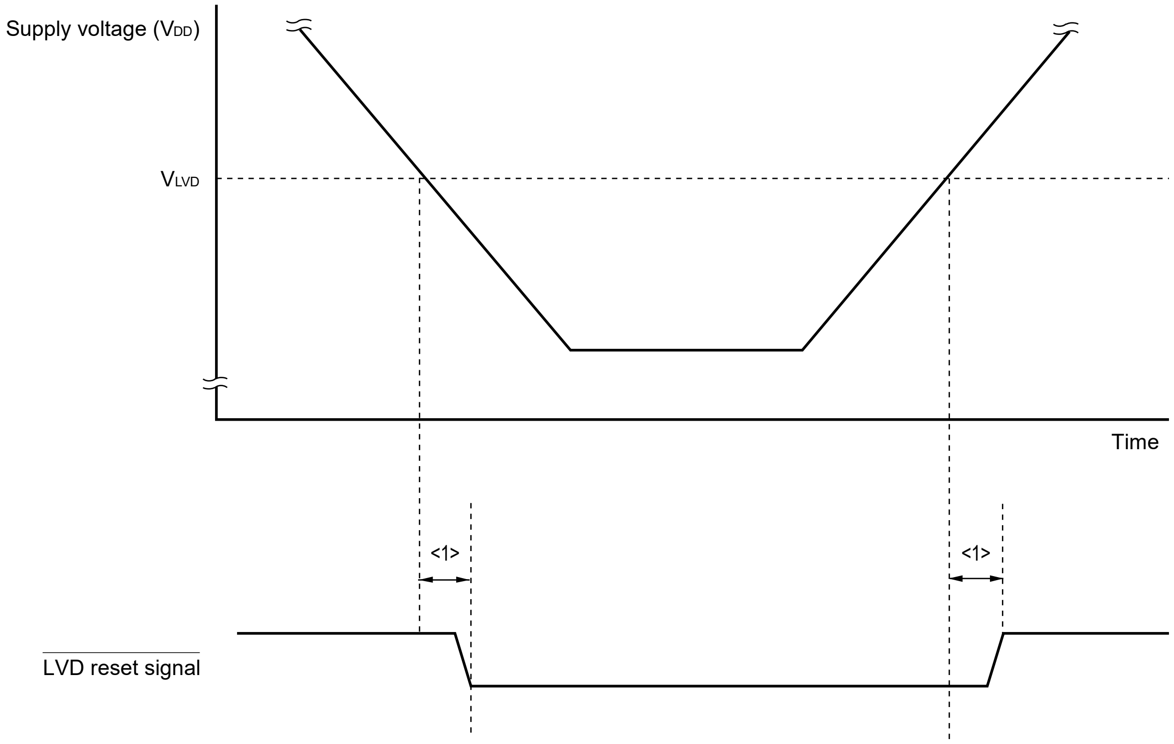
Remark m = 0
n = 0 to 3, 6, 7

(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 19-11**).

Figure 19-11. Delay from the time LVD reset source is generated until the time LVD reset has been generated or released



<1>: Detection delay (300 μs (MAX.))

(3) Power on when LVD is off

Use the external rest input via the $\overline{\text{RESET}}$ pin when the LVD is off.

For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin. To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for 10 μs or more within the operating voltage range shown in **27.4 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 **27.4 AC characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

CHAPTER 20 SAFETY FUNCTIONS

20.1 Overview of Safety Functions

<R> The following safety functions are provided in the R7F0C001G/L, R7F0C002G/L to comply with the IEC60730 safety standards.

These functions enable the microcontroller to self-diagnose abnormalities and stop operating if an abnormality is detected.

(1) Flash memory CRC operation function (high-speed CRC, general-purpose CRC)

This detects data errors in the flash memory by performing CRC operations.

Two CRC functions are provided in the R7F0C001G/L, R7F0C002G/L that can be used according to the application or purpose of use.

- High-speed CRC: The CPU can be stopped and a high-speed check executed on its entire code flash memory area during the initialization routine.
- General CRC: This can be used for checking various data in addition to the code flash memory area while the CPU is running.

(2) RAM parity error detection function

This detects parity errors when reading RAM data.

(3) RAM guard function

This prevents RAM data from being rewritten when the CPU freezes.

(4) SFR guard function

This prevents SFRs from being rewritten when the CPU freezes.

(5) Invalid memory access detection function

This detects illegal accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

(6) Frequency detection function

This function allows a self-check of the CPU/peripheral hardware clock frequencies using the timer array unit.

(7) A/D test function

This is used to perform a self-check of the A/D converter by performing A/D conversion of the A/D converter’s positive and negative reference voltages, analog input channel (ANI), temperature sensor output voltage, and internal reference voltage.

Remark For usage examples of the safety functions complying with the IEC60730 safety standards, refer to RL78 MCU series IEC60730/60335 self-test library application notes (R01AN1062 and R01AN1296).

20.2 Registers Used by Safety Functions

The safety functions use the following registers for each function.

Register	Each Function of Safety Function
• Flash memory CRC control register (CRC0CTL) • Flash memory CRC operation result register (PGCRCL)	Flash memory CRC operation function (high-speed CRC)
• CRC input register (CRCIN) • CRC data register (CRCD)	CRC operation function (general-purpose CRC)
• RAM parity error control register (RPECTL)	RAM parity error detection function
• Invalid memory access detection control register (IAWCTL)	RAM guard function
	SFR guard function
	Invalid memory access detection function
• Timer input select register 0 (TIS0)	Frequency detection function
• A/D test register (ADTES)	A/D test function

The content of each register is described in **20.3 Operation of Safety Functions**.

20.3 Operation of Safety Functions

20.3.1 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 R7F0C001G/L, R7F0C002G/L 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.

20.3.1.1 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 20-1. Format of Flash Memory CRC Control Register (CRC0CTL)

Address: F02F0H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	Control of CRC ALU operation
0	Stop the operation.
1	Start the operation according to HALT instruction execution.

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range
0	0	0	0	0	0	0000H to 3FFBH (16 K to 4 bytes)
0	0	0	0	0	1	0000H to 7FFBH (32 K to 4 bytes)
Other than the above						Setting prohibited

Remark Input the expected CRC operation result value to be used for comparison in the lowest 4 bytes of the flash memory. Note that the operation range will thereby be reduced by 4 bytes.

20.3.1.2 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 20-2. Format of Flash Memory CRC Operation Result Register (PGCRCL)

Address: F02F2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8
	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

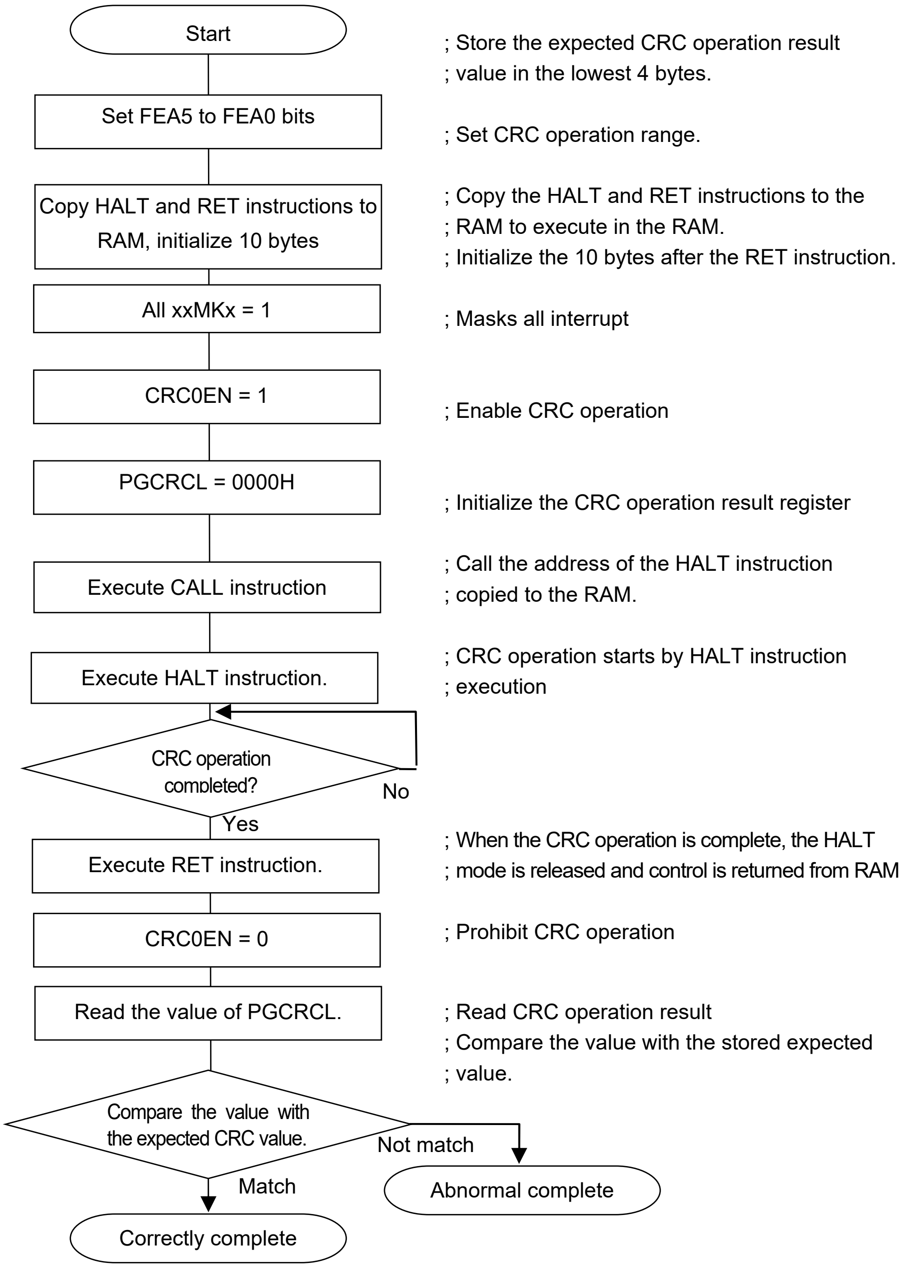
PGCRC15 to PGCRC0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 20-3 shows the flowchart of flash memory CRC operation function (high-speed CRC).

<Operation flow>

Figure 20-3. Flowchart of Flash Memory CRC Operation Function (High-speed CRC)



Cautions 1. The CRC operation is executed only on the code flash.

2. Store the expected CRC operation value in the area below the operation range in the code flash.

3. The CRC operation is enabled by executing the HALT instruction in the RAM area.
Be sure to execute the HALT instruction in RAM area.

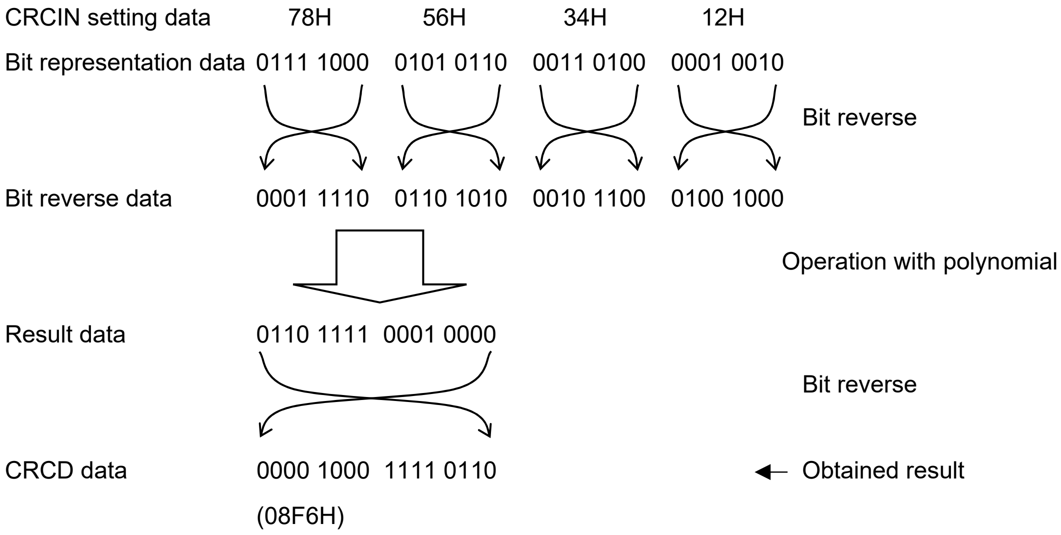
The expected CRC value can be calculated by using the Integrated Development Environment CubeSuite+. See Integrated Development Environment CubeSuite+ user's manual for details.

20.3.2 CRC operation function (general-purpose CRC)

<R> In the R7F0C001G/L, R7F0C002G/L, 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.

20.3.2.1 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 20-4. 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.

20.3.2.2 CRC data register (CRCD)

This register is used to store the CRC operation result of the general-purpose CRC.

The setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register is written, the CRC operation result is stored to the CRCD register.

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

Reset signal generation clears this register to 0000H.

Figure 20-5. Format of CRC Data Register (CRCD)

Address: F02FAH After reset: 0000H R/W

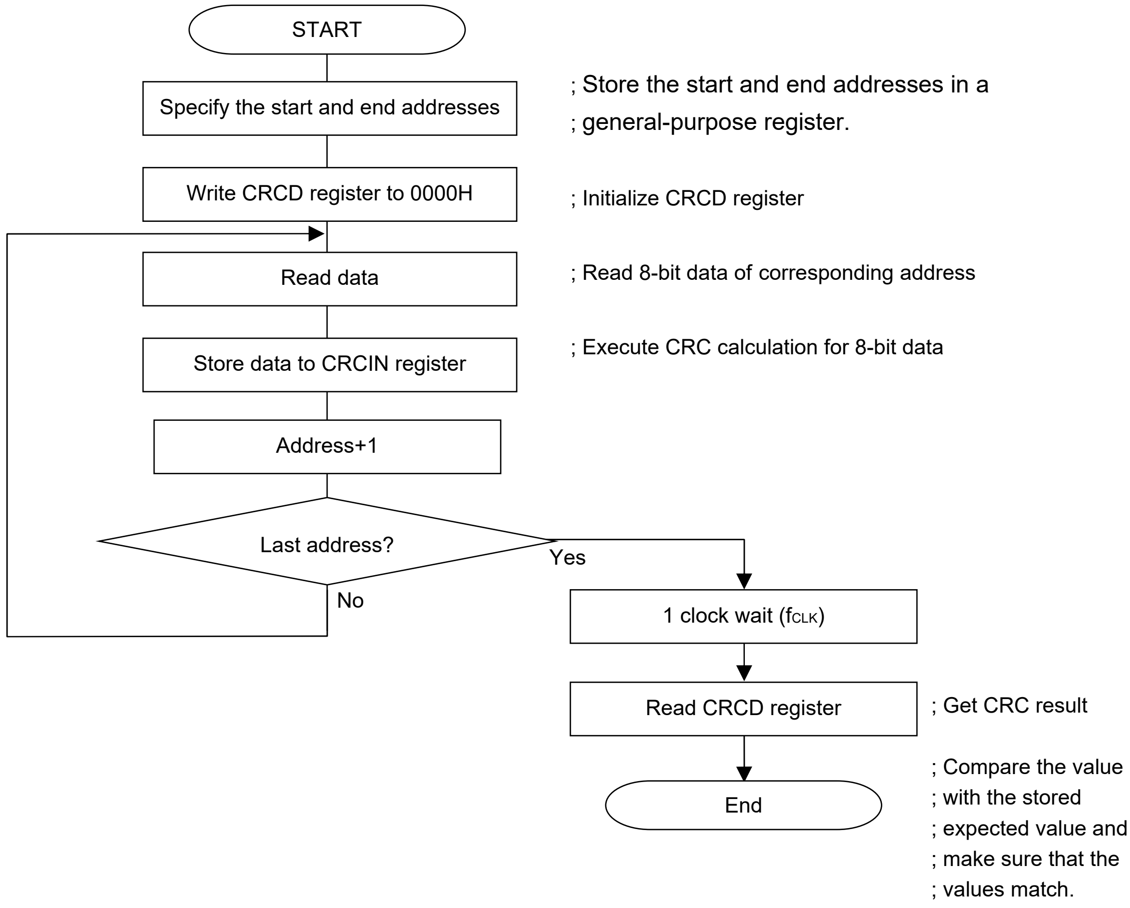
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

Cautions 1. Read the value written to CRCD register before writing to CRCIN register.

2. If conflict between writing and storing operation result to CRCD register occurs, the writing is ignored.

<Operation flow>

Figure 20-6. CRC Operation Function (General-Purpose CRC)



; Store the start and end addresses in a
; general-purpose register.

; Initialize CRCD register

; Read 8-bit data of corresponding address

; Execute CRC calculation for 8-bit data

; Get CRC result

; Compare the value
; with the stored
; expected value and
; make sure that the
; values match.

20.3.3 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 R7F0C001G/L, R7F0C002G/L'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.

20.3.3.1 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 20-7. Format of RAM Parity Error Control Register (RPECTL)

Address: F00F5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

RPERDIS	Parity error reset mask flag
0	Enable parity error resets.
1	Disable parity error resets.

RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

Caution The parity bit is appended when data is written, and the parity is checked when the data is read.

Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed before reading data.

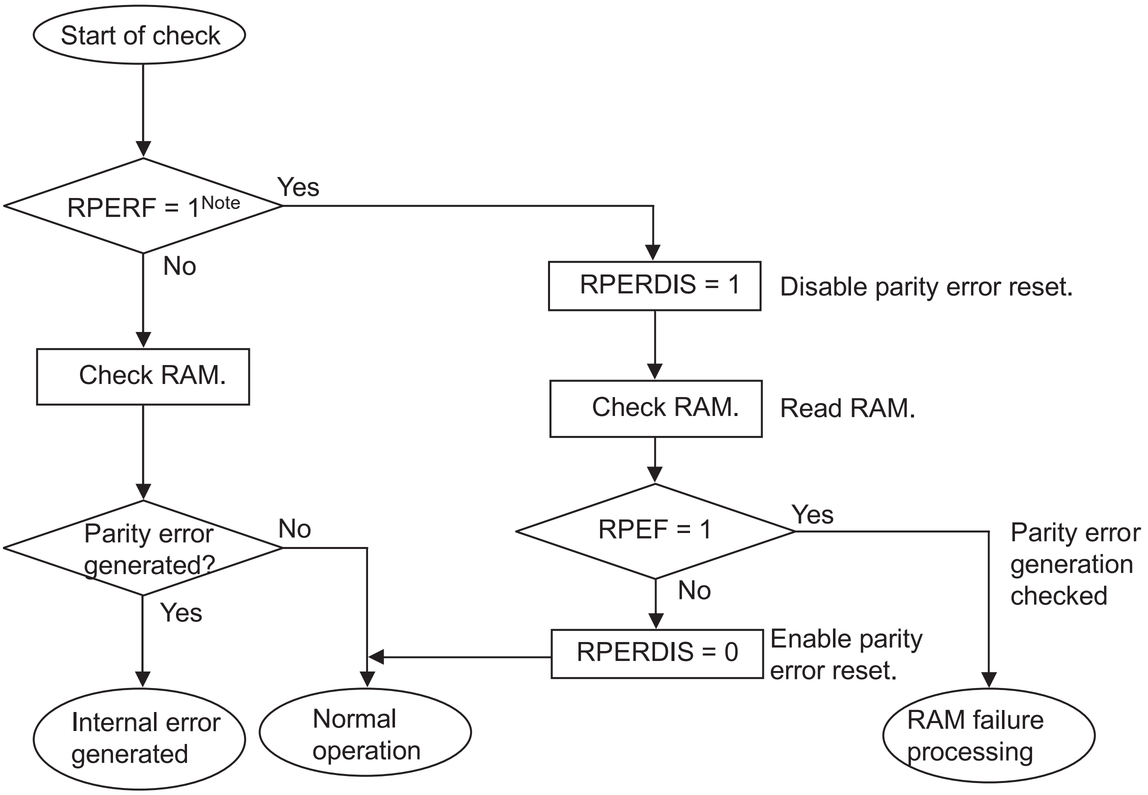
The RL78's CPU executes look-ahead due to the pipeline operation, the CPU might read an uninitialized RAM area that is allocated beyond the RAM used, which causes a RAM parity error.

Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area + 10 bytes when instructions are fetched from RAM areas.

Remarks

1. The parity error reset is enabled by default (RPERDIS = 0).
2. Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set (1) if a parity error occurs. If parity error resets are enabled (RPERDIS = 0) with RPEF set to 1, a parity error reset is generated when the RPERDIS bit is cleared to 0.
3. The RPEF flag in the RPECTL register is set (1) when the RAM parity error occurs and cleared (0) by writing 0 to it or by any reset source. When RPEF = 1, the value is retained even if RAM for which no parity error has occurred is read.
4. The general registers are not included for RAM parity error detection.

Figure 20-8. Flowchart of RAM Parity Check



Note To check internal reset status using a RAM parity error, see **CHAPTER 17 RESET FUNCTION**.

20.3.4 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.

20.3.4.1 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 20-9. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM guard space[Note]
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes starting at the RAM address
1	0	The 256 bytes starting at the RAM address
1	1	The 512 bytes starting at the RAM address

Note The RAM start address differs depending on the size of the RAM provided with the product.

20.3.5 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.

20.3.5.1 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 20-10. 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 1

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 Notes 2	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

Notes **1.** Pxx (Port register) is not guarded.

2. Clear GCSC bit to 0, during self programming /serial programming.

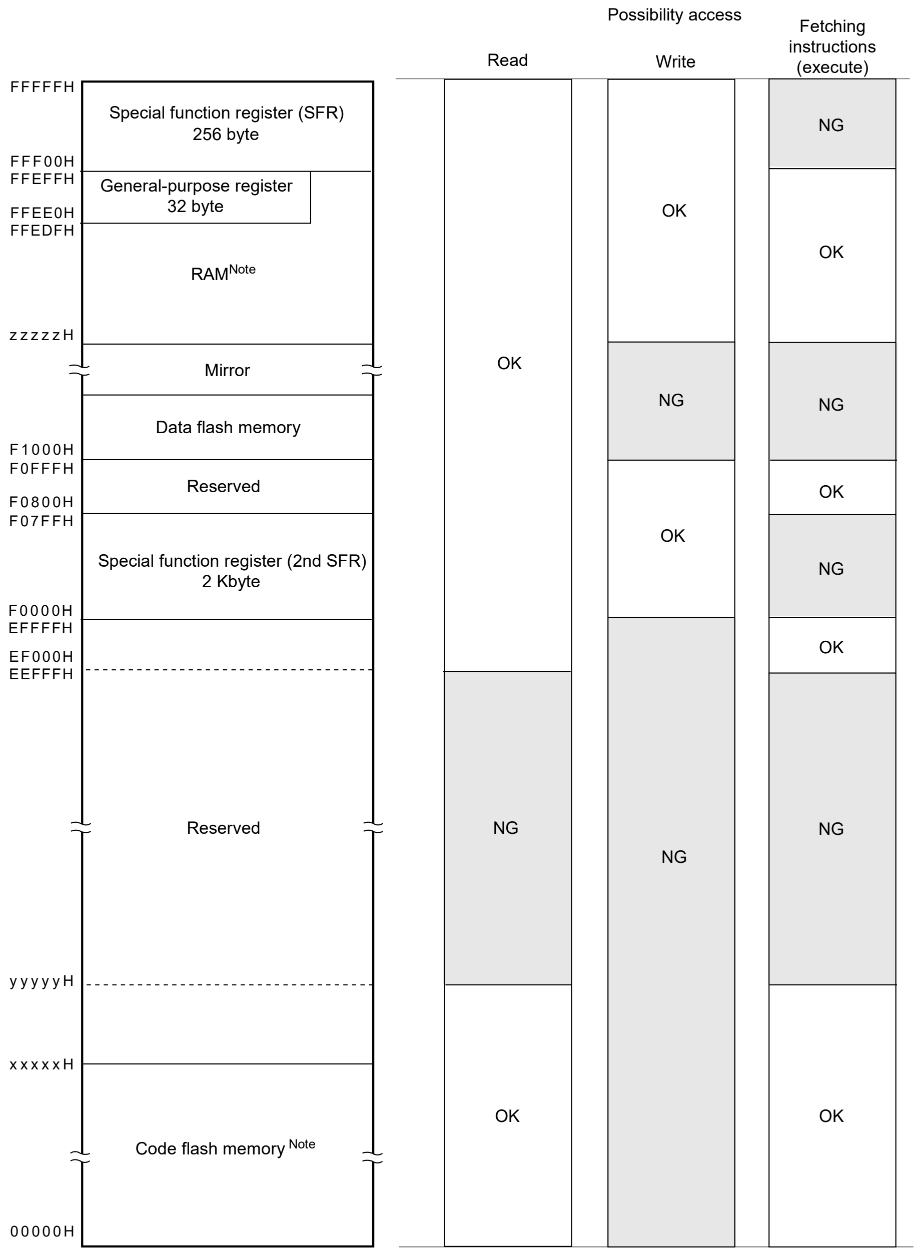
20.3.6 Invalid memory access detection function

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 20-11.

Figure 20-11. 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)
R7F0C001G, R7F0C001L	16384 × 8 bit (00000H to 03FFFH)	1024 × 8 bit (FFB00H to FFEFFH)	10000H
R7F0C002G, R7F0C002L	32768 × 8 bit (00000H to 07FFFH)	1536 × 8 bit (FF900H to FFEFFH)	10000H

20.3.6.1 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 20-12. 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	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 for the option byte, the invalid memory access function is always enabled regardless of the setting for the IAWEN bit. (For details, see **CHAPTER 22 OPTION BYTE**.)

20.3.7 Frequency detection function

The IEC60730 standard mandates checking that the oscillation frequency is correct.

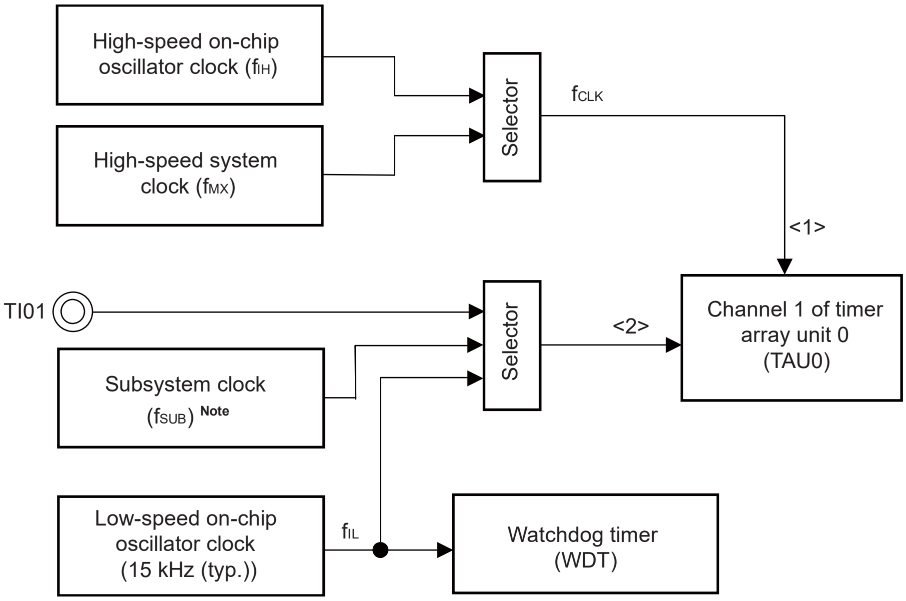
By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring the pulse width of the input signal to channel 1 of the timer array unit 0 (TAU0), whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

<Clocks to be compared>

<1> CPU/peripheral hardware clock frequency (f_{CLK}):
- High-speed on-chip oscillator clock (f_{IH})
- High-speed system clock (f_{MX})

<2> Input to channel 1 of the timer array unit
- Timer input to channel 1 (TI01)
- Low-speed on-chip oscillator clock (f_{IL}: 15 kHz (typ.))
- Subsystem clock (f_{SUB}) **Note**

Figure 20-13. Configuration of Frequency Detection Function



<Operational overview>

Whether the clock frequency is correct or not can be judged by measuring the pulse interval under the following conditions:

- The high-speed on-chip oscillator clock (f_{IH}) or the external X1 oscillation clock (f_{MX}) is selected as the CPU/peripheral hardware clock (f_{CLK}).
- The low-speed on-chip oscillator clock (f_{IL}: 15 kHz) is selected as the timer input for channel 1 of timer array unit 0 (TAU0).

If input pulse interval measurement results in an abnormal value, it can be concluded that the clock frequency is abnormal.

For how to execute input pulse interval measurement, see **6.8.4 Operation as input pulse interval measurement**.

Note Can only be selected in the products incorporating the subsystem clock.

20.3.7.1 Timer input select register 0 (TIS0)

This register is used to select the timer input of channel 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 20-14. 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	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than the above			Setting prohibited

20.3.8 A/D test function

The IEC60730 standard mandates testing the A/D converter. The A/D test function checks whether or not the A/D converter is operating normally by executing A/D conversions of the A/D converter's positive and negative reference voltages, analog input channel (ANI), temperature sensor output voltage, and the internal reference voltage. For details of the check method, see safety function (A/D test) application note (R01AN0955).

The analog multiplexer can be checked using the following procedure.

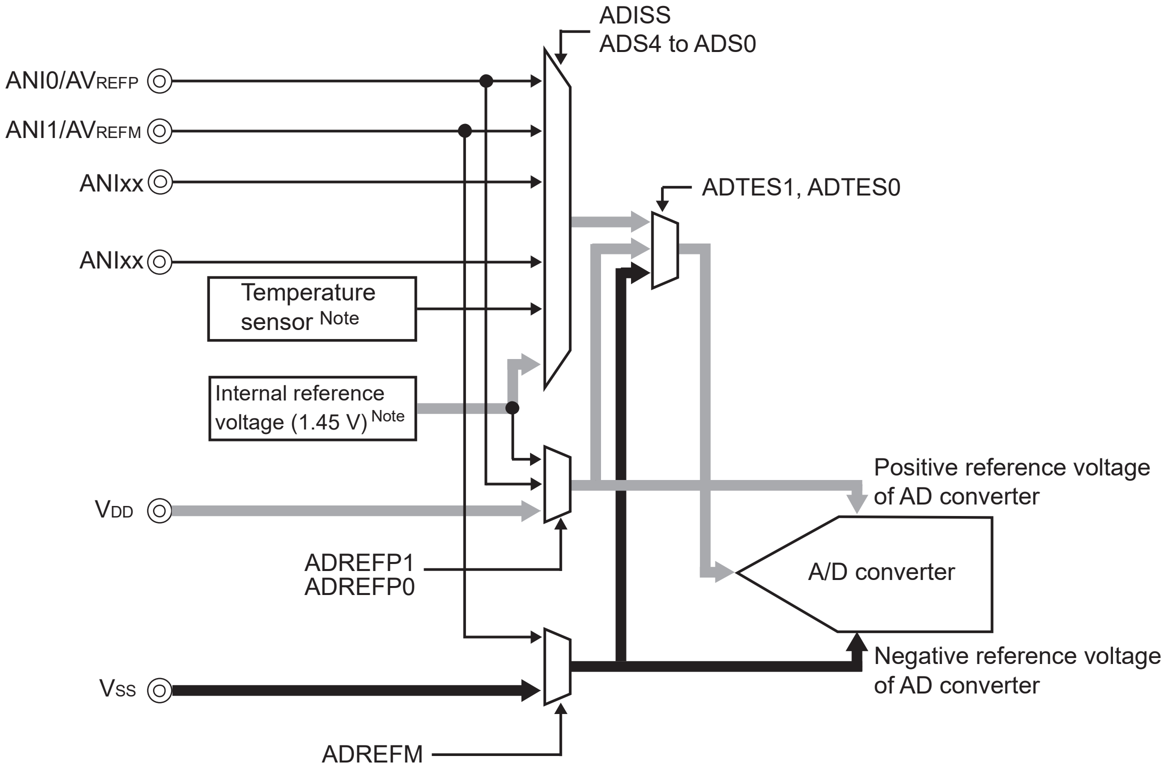
<1> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
<2> Perform A/D conversion for the ANIx pin (conversion result 1-1).
<3> Select the A/D converter's negative reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 0)
<4> Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
<5> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
<6> Perform A/D conversion for the ANIx pin (conversion result 1-2).
<7> Select the A/D converter's positive reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 1)
<8> Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
<9> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
<10> Perform A/D conversion for the ANIx pin (conversion result 1-3).
<11> Check that the conversion results 1-1, 1-2, and 1-3 are equal.
<12> Check that the A/D conversion result 2-1 is all zero and conversion result 2-2 is all one.

Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Remarks 1. If the analog input voltage is variable during A/D conversion in steps <1> to <10> above, use another method to check the analog multiplexer.

2. The conversion results might contain an error. Consider an appropriate level of error when comparing the conversion results.

Figure 20-15. Configuration of A/D Test Function



Note This setting can be used only in HS (high-speed main) mode.

20.3.8.1 A/D test register (ADTES)

This register is used to select the A/D converter’s positive reference voltage, A/D converter’s negative reference voltage, analog input channel (ANIxx), temperature sensor output voltage, or internal reference voltage (1.45 V) as the target of A/D conversion.

When using the A/D test function, specify the following settings:

- Select negative reference voltage as the target of A/D conversion for zero-scale measurement.
- Select positive reference voltage as the target of A/D conversion for full-scale measurement.

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

Reset signal generation clears this register to 00H.

Figure 20-16. 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 with the ADREFM bit in ADM2)
1	1	Positive reference voltage (selected with the ADREFP1 or ADREFP0 bit in ADM2)
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.

20.3.8.2 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 20-17. 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	P20/ANI0/AV_{REFP} pin
0	0	0	0	0	1	ANI1	P21/ANI1/AV_{REFM} pin
0	1	0	0	0	0	ANI16	P41/ANI16 pin
0	1	0	0	0	1	ANI17	P120/ANI17 pin
0	1	0	0	1	0	ANI18	P13/ANI18 pin
0	1	0	0	1	1	ANI19	P14/ANI19 pin
0	1	0	1	0	0	ANI20	P142/ANI20 pin
0	1	0	1	0	1	ANI21	P143/ANI21 pin
0	1	0	1	1	0	ANI22	P144/ANI22 pin
0	1	0	1	1	1	ANI23	P145/ANI23 pin
1	0	0	0	0	0	–	Temperature sensor output voltage [Note]
1	0	0	0	0	1	–	Internal reference voltage (1.45 V) [Note]
Other than the above						Setting prohibited	

Note Can only be used in HS (high-speed main) mode.

Cautions

1. **Be sure to clear bits 5 and 6 to 0.**
2. **Select input mode for the ports which are set to analog input with the ADPC and PMC registers, using the port mode registers 1, 2, 4, 12, and 14 (PM1, PM2, PM4, PM12, and PM14).**
3. **Do not use the ADS register to set the pins which should be set as digital I/O with the A/D port configuration register (ADPC).**
4. **Do not use the ADS register to set the pins which should be set as digital I/O with the port mode control registers 1, 4, 12, and 14 (PMC1, PMC4, PMC12, and PMC14).**
5. **Only rewrite the value of the ADISS bit while conversion operation is stopped (ADCS = 0, ADCE = 0).**
6. **If using AV_{REFP} as the + side reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.**
7. **If using AV_{REFM} as the – side reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.**
8. **When ADISS is 1, the internal reference voltage (1.45 V) cannot be used for the positive reference voltage. In addition, the first conversion result obtained after setting ADISS to 1 is not available. For detailed setting flow, see 11.7.4 Setup when temperature sensor**

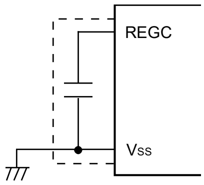
output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode).

9. **If a transition is made to STOP mode or a transition is made to HALT mode during CPU operation with subsystem clock, do not set ADISS to 1. When ADISS is 1, the A/D converter reference voltage current (I_{ADREF}) shown in 27.3.2 Supply current characteristics is added.**

CHAPTER 21 REGULATOR

21.1 Regulator Overview

The R7F0C001G/L, R7F0C002G/L contain a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

The regulator output voltage, see Table 21-1.

Table 21-1. 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 (f_{MX}) and the high-speed on-chip oscillator clock (f_{IH}) are stopped during CPU operation with the subsystem clock (f_{XT})
		When both the high-speed system clock (f_{MX}) and the high-speed on-chip oscillator clock (f_{IH}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{XT}) 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 22 OPTION BYTE

22.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the R7F0C001G/L, R7F0C002G/L 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, be sure to set the value specified in this manual.

22.1.1 User option byte (000C0H to 000C2H)

(1) 000C0H

- O Operation of watchdog timer
  - ● Operation is stopped or enabled in the HALT or STOP mode.
- O Setting of overflow time of watchdog timer
- O Operation of watchdog timer
  - ● Operation is stopped or enabled.
- O Setting of window open period of watchdog timer
- O Setting of interval interrupt of watchdog timer
  - ● Used or not used

(2) 000C1H

- O Setting of LVD operation mode
  - ● Interrupt & reset mode.
  - ● Reset mode.
  - ● Interrupt mode.
  - ● LVD off (by controlling the externally input reset signal on the $\overline{\text{RESET}}$ pin)
- O Setting of LVD detection level (V_{LVDH}, V_{LVDL}, V_{LVD})

Caution After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 27.4 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).

(3) 000C2H

- O 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
- O Setting of the frequency of the high-speed on-chip oscillator
  - ● Select from 24 MHz/16 MHz/12 MHz/8 MHz/6 MHz/4 MHz/3 MHz/2 MHz/1 MHz (TYP.).

22.1.2 On-chip debug option byte (000C3H)

O Control of on-chip debug operation

- On-chip debug operation is disabled or enabled.

O Handling of data of flash memory in case of failure in on-chip debug security ID authentication

- Data of flash memory is erased or not erased in case of failure in on-chip debug security ID authentication.

22.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 22-1. Format of User Option Byte (000C0H)

Address: 000C0H

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2f_{IL} of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period Note 1
0	0	Setting prohibited
0	1	50%
1	0	75% Note 2
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.19 ms)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode Note 1
1	Counter operation enabled in HALT/STOP mode

Notes 1. The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

2. 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 **The watchdog timer continues its operation even during self-programming or data flash rewrite. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.**

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 22-2. Format of User Option Byte (000C1H) (1/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- 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		
3.75 V	3.67 V					0	0		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
Other than above			Setting prohibited						

- 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		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
Other than above		Setting prohibited						

Remarks 1. For details on the LVD circuit, see **CHAPTER 19 VOLTAGE DETECTOR**.

2. The detection voltage is a typical value. For details, see **27.6.4 LVD circuit characteristics**.

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

Figure 22-2. Format of User Option Byte (000C1H) (2/2)

Address: 000C1H

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
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		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
Other than above		Setting 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	×	×	×	×	0/1	1
Other than above		Setting prohibited						

Cautions 1. Be sure to set bit 4 to “1”.

2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 27.4 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).

Remarks 1. ×: don’t care

2. For details on the LVD circuit, see **CHAPTER 19 VOLTAGE DETECTOR**.

3. The detection voltage is a typical value. For details, see **27.6.4 LVD circuit characteristics**.

Figure 22-3. Format of Option Byte (000C2H)

Address: 000C2H

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

<table>
<tr><td rowspan="2">CMODE1</td><td rowspan="2">CMODE0</td><td colspan="3">Setting of flash operation mode</td></tr>
<tr><td></td><td>Operating frequency range (f_{MAIN})</td><td>Operating voltage range (V_{DD})</td></tr>
<tr><td>0</td><td>0</td><td>LV (low voltage main) mode</td><td>1 to 4 MHz</td><td>1.6 to 5.5 V</td></tr>
<tr><td>1</td><td>0</td><td>LS (low speed main) mode</td><td>1 to 8 MHz</td><td>1.8 to 5.5 V</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">1</td><td rowspan="2">HS (high speed main) mode</td><td>1 to 16 MHz</td><td>2.4 to 5.5 V</td></tr>
<tr><td>1 to 24 MHz</td><td>2.7 to 5.5 V</td></tr>
<tr><td colspan="2">Other than above</td><td colspan="3">Setting prohibited</td></tr>
</table>

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

Cautions 1. Be sure to set 10B to bits 5 and 4.

2. The ranges of operating frequency and operating voltage vary depending on the flash operation mode. For details, see 27.4 AC Characteristics.

22.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 22-4. Format of On-chip Debug Option Byte (000C3H)

Address: 000C3H

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Enables on-chip debugging. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debugging. Does not erase data of flash memory in case of failures in authenticating on-chip debug security ID.

Caution Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value.
Be sure to set 000010B to bits 6 to 1.

Remark The value on bits 3 to 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.
However, be sure to set the default values (0, 1, and 0) to bits 3 to 1 at setting.

22.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the assembler linker option, in addition to describing to the source. When doing so, the contents set by using the linker option take precedence, even if descriptions exist in the source, as mentioned below.

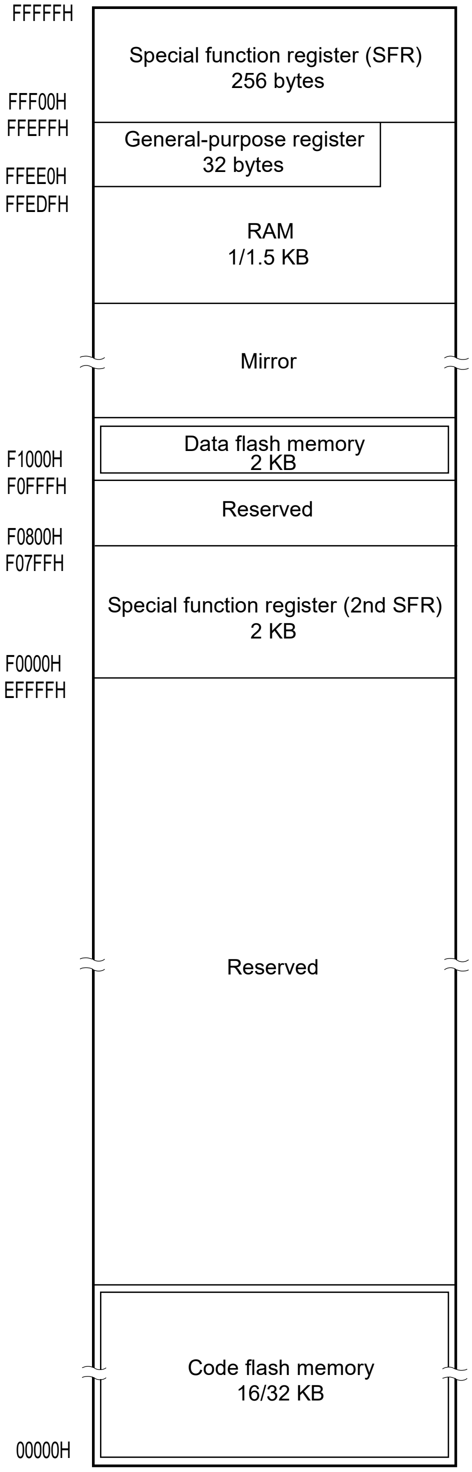
A software description example of the option byte setting is shown below.

```
OPT   CSEG          OPT_BYTE
      DB            36H       ;  Does not use interval interrupt of watchdog timer,
                              ;  Enables watchdog timer operation,
                              ;  Window open period of watchdog timer is 50%,
                              ;  Overflow time of watchdog timer is 2^9/fIL,
                              ;  Stops watchdog timer operation during HALT/STOP mode
      DB            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
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction.

CHAPTER 23 FLASH MEMORY

The R7F0C001G/L, R7F0C002G/L 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 **23.1**)
  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.

- Serial programming using external device (UART communication) (see **23.2**)
  Data can be written to the flash memory on-board through UART communication with an external device (microcontroller or ASIC).

- Self-programming (see **23.6**)
  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 **23.8 Data Flash**.

23.1 Writing to Flash Memory by Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the R7F0C001G/L, R7F0C002G/L.

- PG-FP5, FL-PR5
- E1 on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the R7F0C001G/L, R7F0C002G/L 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 R7F0C001G/L, R7F0C002G/L is mounted on the target system.

Remark FL-PR5 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

Table 23-1. Wiring Between R7F0C001G/L, R7F0C002G/L and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.	
					48-pin	64-pin
Signal Name		I/O	Pin Function		LQFP (7x7)	LQFP (10x10)
PG-FP5, FL-PR5	E1 on-chip debugging emulator					
–	TOOL0	I/O	Transmit/receive signal	TOOL0/ P40	3	5
SI/RxD	–	I/O	Transmit/receive signal			
SCK	–	Output	–	–	–	–
CLK	–	Output	–	–	–	–
–	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	4	6
/RESET	–	Output				
FLMD0	–	Output	Mode signal	–	–	–
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	12	15
GND		–	Ground	V_{SS}	11	13
				EV_{SS}	–	14
				REGC **Note**	10	12
FLMD1	EMV_{DD}	–	Driving power for TOOL0 pin	V_{DD}	12	–
				EV_{DD}	–	16

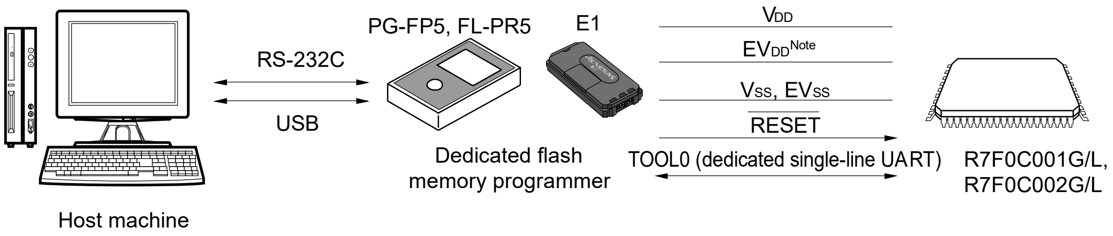
Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

23.1.1 Programming Environment

The environment required for writing a program to the flash memory of the R7F0C001G/L, R7F0C002G/L is illustrated below.

Figure 23-1. Environment for Writing Program to Flash Memory



Note 64-pin products only.

A host machine that controls the dedicated flash memory programmer is necessary.

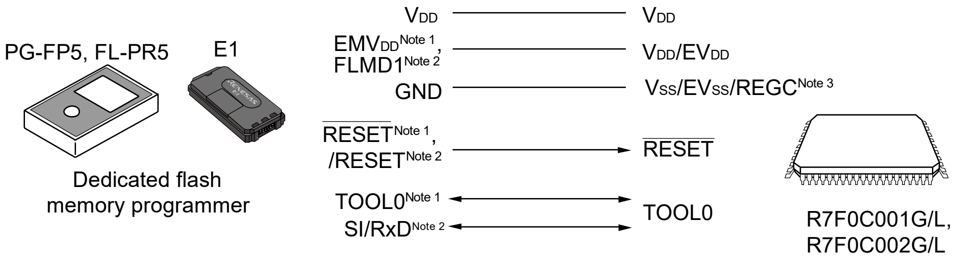
To interface between the dedicated flash memory programmer and the R7F0C001G/L, R7F0C002G/L, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

23.1.2 Communication Mode

Communication between the dedicated flash memory programmer and the R7F0C001G/L, R7F0C002G/L is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the R7F0C001G/L, R7F0C002G/L.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 23-2. Communication with Dedicated Flash Memory Programmer



Notes **1.** When using E1 on-chip debugging emulator.
2. When using PG-FP5 or FL-PR5.
3. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The dedicated flash memory programmer generates the following signals for the R7F0C001G/L, R7F0C002G/L. See the manual of PG-FP5, FL-PR5, or E1 on-chip debugging emulator for details.

Table 23-2. Pin Connection

Dedicated Flash Memory Programmer				R7F0C001G/L, R7F0C002G/L
Signal Name		I/O	Pin Function	Pin Name
PG-FP5, FL-PR5	E1 on-chip debugging emulator			
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		–	Ground	V_{SS}, EV_{SS}, REGC [Note]
FLMD1	EMV_{DD}	–	Driving power for TOOL0 pin	V_{DD}, EV_{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 Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Caution Pins to be connected differ with the product. For details, see Table 23-1.

23.2 Writing to Flash Memory by Using External Device (that Incorporates UART)

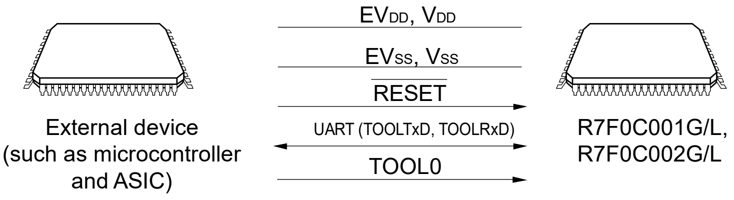
On-board data writing to the internal flash memory is possible by using the R7F0C001G/L, R7F0C002G/L and an external device (a microcontroller or ASIC) connected to a UART.

On the development of flash memory programmer by user, refer to RL78 microcontroller (RL78 Protocol A) Programmer Edition Application Note (R01AN0815).

23.2.1 Programming Environment

The environment required for writing a program to the flash memory of the R7F0C001G/L, R7F0C002G/L is illustrated below.

Figure 23-3. Environment for Writing Program to Flash Memory



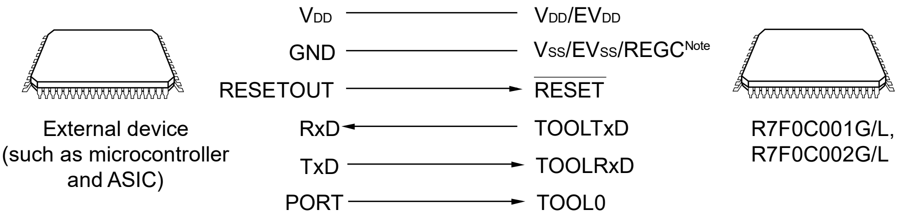
Processing to write data to or delete data from the R7F0C001G/L, R7F0C002G/L by using an external device is performed on-board. Off-board writing is not possible.

23.2.2 Communication Mode

Communication between the external device and the R7F0C001G/L, R7F0C002G/L is established by serial communication using the TOOLTxD and TOOLRxD pins via the dedicated UART of the R7F0C001G/L, R7F0C002G/L.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 23-4. 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 R7F0C001G/L, R7F0C002G/L.

Table 23-3. Pin Connection

External Device			R7F0C001G/L, R7F0C002G/L
Signal Name	I/O	Pin Function	Pin Name
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, EV_{DD}
GND	–	Ground	V_{SS}, EV_{SS}, 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).

23.3 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 flash programming mode, see **23.6 Self-Programming**.

23.3.1 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 1 ms period after external pin reset release. Furthermore, when this pin is used via pull-down resistors, use the 500 kΩ or more resistors.
When used as an output pin:	When this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

Remarks 1. t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end for setting of the flash memory programming mode (see **27.11 Timing Specifications for Switching Flash Memory Programming Modes**).

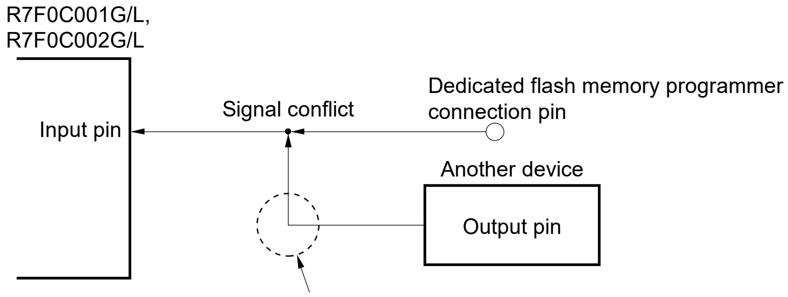
2. The SAU and IICA pins are not used for communication between the R7F0C001G/L, R7F0C002G/L and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

23.3.2 $\overline{\text{RESET}}$ 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 23-5. Signal Conflict ($\overline{\text{RESET}}$ Pin)



In the flash memory programming mode, a signal output by another device will conflict with the signal output by the dedicated flash memory programmer. Therefore, isolate the signal of another device.

23.3.3 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 EV_{DD}, or V_{SS} or EV_{SS}, via a resistor.

23.3.4 REGC pin

Connect the REGC pin to GND via a capacitor (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.

23.3.5 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the high-speed on-chip oscillator clock (f_{IH}) is used.

23.3.6 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.

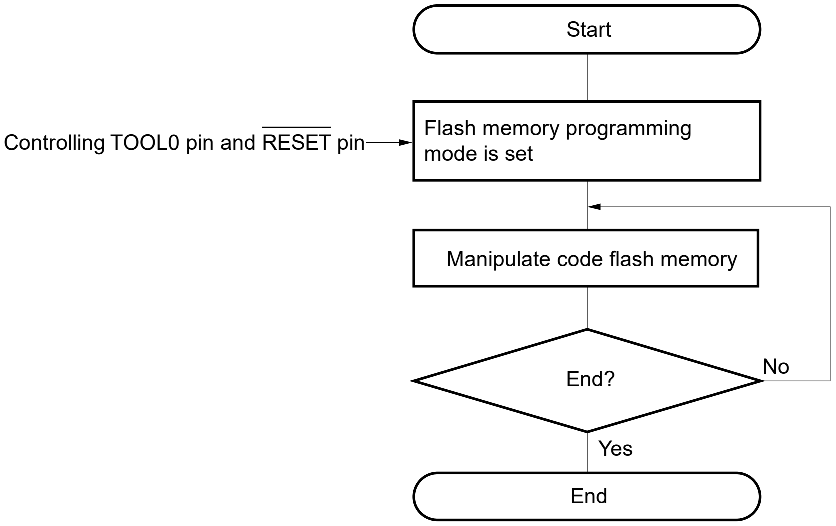
Supply the same other power supplies (EV_{DD}, EV_{SS}) as those V_{DD} and V_{SS}.

23.4 Serial Programming Method

23.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 23-6. Code Flash Memory Manipulation Procedure



23.4.2 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 R7F0C001G/L, R7F0C002G/L to a dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

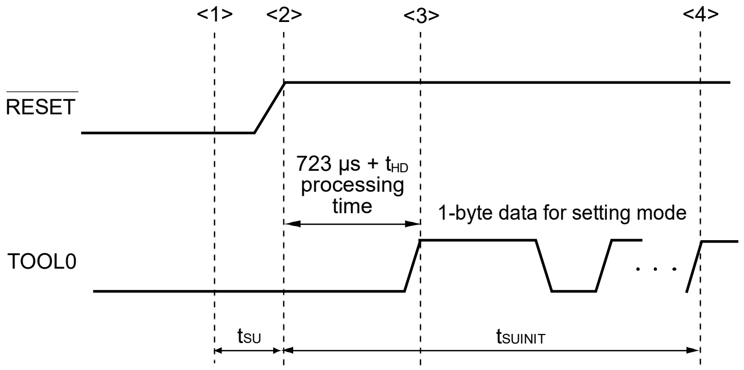
<Serial programming using an external device (UART communication)>

Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 23-4**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in **Figure 23-7**. For details, refer to **R7F0C001G/L, R7F0C002G/L (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 23-4. Relationship between TOOL0 Pin and Operation Mode after Reset Release

TOOL0	Operation Mode
EV_{DD}	Normal operation mode
0 V	Flash memory programming mode

Figure 23-7. Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Baud rate setting by UART reception is completed.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the resets end.

t_{SU}: How long from when the TOOL0 pin is placed at the low level until an external reset ends

t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end (the flash firmware processing time is excluded)

For details, see **27.11 Timing Specifications for Switching 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 23-5. 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	
$2.7\ V \leq V_{DD} \leq 5.5\ V$	Blank state		Full speed mode
	HS (high speed main) mode	1 MHz to 32 MHz	Full speed mode
	LS (low speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low voltage main) mode	1 MHz to 4 MHz	Wide voltage mode
$2.4\ V \leq V_{DD} < 2.7\ V$	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
$1.8\ V \leq V_{DD} < 2.4\ V$	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

Remarks 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.

2. For details about communication commands, see **23.4.4 Communication commands**.

23.4.3 Selecting communication mode

Communication modes of the R7F0C001G/L, R7F0C002G/L are as follows.

Table 23-6. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed [Note 2]	Frequency	Multiply Rate	
1-line UART (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

Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

23.4.4 Communication commands

The R7F0C001G/L, R7F0C002G/L executes serial programming through the commands listed in **Table 23-7.**

The signals sent from the dedicated flash memory programmer or external device to the R7F0C001G/L, R7F0C002G/L are called commands, and programming functions corresponding to the commands are executed. For details, refer to **RL78 microcontroller (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 23-7. 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 R7F0C001G/L, R7F0C002G/L 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.

Table 23-8 is a list of signature data and **Table 23-9** shows an example of signature data.

Table 23-8. 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 07FFFH (32 KB) → FFH, 7FH, 00H)	3 bytes
Data flash memory area last address	Last address of data flash memory area (Sent from lower address. Example: F1000H to F17FFH (2 KB) → FFH, 17H, 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 23-9. Example of Signature Data

Field name	Description	Number of transmit data	Data (hexadecimal)
Device code	RL78 protocol A	3 bytes	10 00 06
Device name	R7F0C002L	10 bytes	52 = “R” 37 = “7” 46 = “F” 30 = “0” 43 = “C” 30 = "0" 30 = "0" 32 = “2” 4C = “L” 20 = “ ”
Code flash memory area last address	Code flash memory area 00000H to 07FFFH (32 KB)	3 bytes	FF 7F 00
Data flash memory area last address	Data flash memory area F1000H to F17FFH (2 KB)	3 bytes	FF 17 0F
Firmware version	Ver.1.23	3 bytes	01 02 03

23.5 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP5 is used as a dedicated flash memory programmer.

Table 23-10. Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

PG-FP5 Command	Code Flash	
	16 Kbytes	32 Kbytes
Erasing	1.5 s	1.5 s
Writing	1.5 s	2 s
Verification	1.5 s	1.5 s
Writing after erasing	2 s	2.5 s

Remark The command processing times (reference values) shown in the table are typical values under the following conditions.

Port: TOOL0 (single-line UART)

Speed: 1,000,000 bps

Mode: Full speed mode (flash operation mode: HS (high speed main) mode)

23.6 Self-Programming

The R7F0C001G/L, R7F0C002G/L supports a self-programming function that can be used to rewrite the code flash memory via a user program. Because this function allows a user application to rewrite the code flash memory by using the flash self-programming library, it can be used to upgrade the program in the field.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.

2. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the flash self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the flash self-programming library.

3. The high-speed on-chip oscillator should be kept operating during self-programming. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The flash self-programming library should be executed after 30 μs have elapsed.

Remarks 1. For details of the self-programming function, refer to **RL78 Microcontroller Flash Self Programming Library Type01 User’s Manual (R01US0050).**

2. For details of the time required to execute self-programming, see the notes on use that accompany the flash self-programming library tool.

The self-programming function has two flash memory programming modes; wide voltage mode and full speed mode.

Specify the mode that corresponds to the flash operation mode specified in bits CMODE1 and CMODE0 in option byte 000C2H.

Specify the full speed mode when the HS (high speed main) mode is specified. Specify the wide voltage mode when the LS (low speed main) mode or LV (low voltage main) mode is specified.

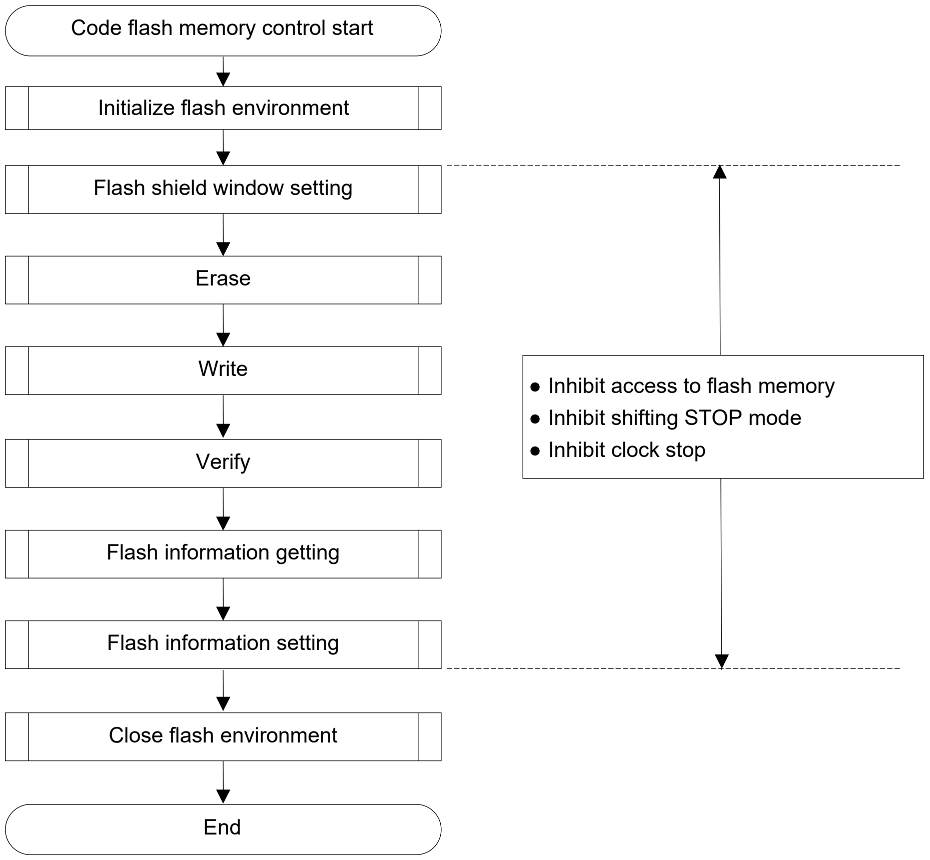
If the argument fsl_flash_voltage_u08 is 00H when the FSL_Init function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

Remark Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.

23.6.1 Self-programming procedure

The following figure illustrates a flow for rewriting the code flash memory by using a flash self-programming library.

Figure 23-8. Flow of Self Programming (Rewriting Flash Memory)



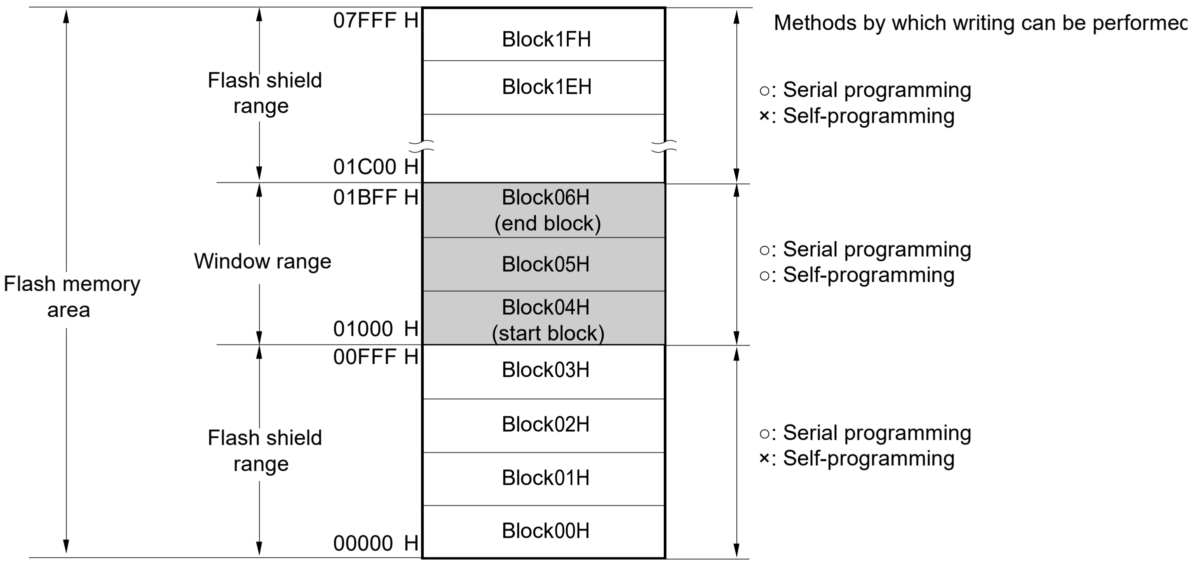
23.6.2 Flash shield window function

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 23-9. Flash Shield Window Setting Example
(Target Devices: R7F0C002G, R7F0C002L, Start Block: 04H, End Block: 06H)



Cautions 1. If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.

2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).

Table 23-11. 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 **23.7 Security Settings** to prohibit writing/erasing during serial programming.

23.7 Security Settings

The R7F0C001G/L, R7F0C002G/L supports a security function that prohibits rewriting the user program written to the code 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 code 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 23-12 shows the relationship between the erase and write commands when the R7F0C001G/L, R7F0C002G/L 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 **23.6.2** for detail).

Table 23-12. 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 **23.6.2** for detail).

Table 23-13. 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) During self-programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set by using flash self-programming library.	Cannot be disabled after set.
Prohibition of writing		Cannot be disabled during self-programming (set via GUI of dedicated flash memory programmer, etc. during serial programming).
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

23.8 Data Flash

23.8.1 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.

Cautions 1. The data flash memory is stopped after a reset is canceled. The data flash control register (DFLCTL) must be set up in order to use the data flash memory.

2. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The data flash library should be executed after 30 μs have elapsed.

Remark For rewriting the code flash memory via a user program, see **23.6 Self-Programming**.

23.8.2 Register controlling data flash memory

23.8.2.1 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 23-10. Format of Data Flash Control Register (DFLCTL)

Address: F0090H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	Data flash access control
0	Disables data flash access
1	Enables data flash access

Caution Manipulating the DFLCTL register is not possible while rewriting the data flash memory.

23.8.3 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.

<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.

Cautions 1. Accessing the data flash memory is not possible during the setup time.

2. Transition to the STOP mode is not possible during the setup time. To enter the STOP mode during the setup time, clear DFLEN to 0 and then execute the STOP instruction.

3. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The data flash library should be executed after 30 μs have elapsed.

<R> **4. Once the data flash memory is read while the subsystem clock is selected as the CPU/peripheral hardware clock (CLS = 1), follow the procedure listed as steps (1) to (3) below, in that order, to read the data flash area after switching the CPU/peripheral hardware clock from the subsystem clock to the main system clock.**

(1) Make sure the main system clock is selected as the CPU/peripheral hardware clock (CLS = 0).

(2) Read data from any location in the data flash area. The value read at this point is undefined.

(3) Wait for the time listed below according to the operating mode, then read data from the desired parts of the data flash area.

HS (high-speed main) mode: 5 μs
LS (low-speed main) mode: 1 μs
LV (low-voltage main) mode: 10 μs

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.

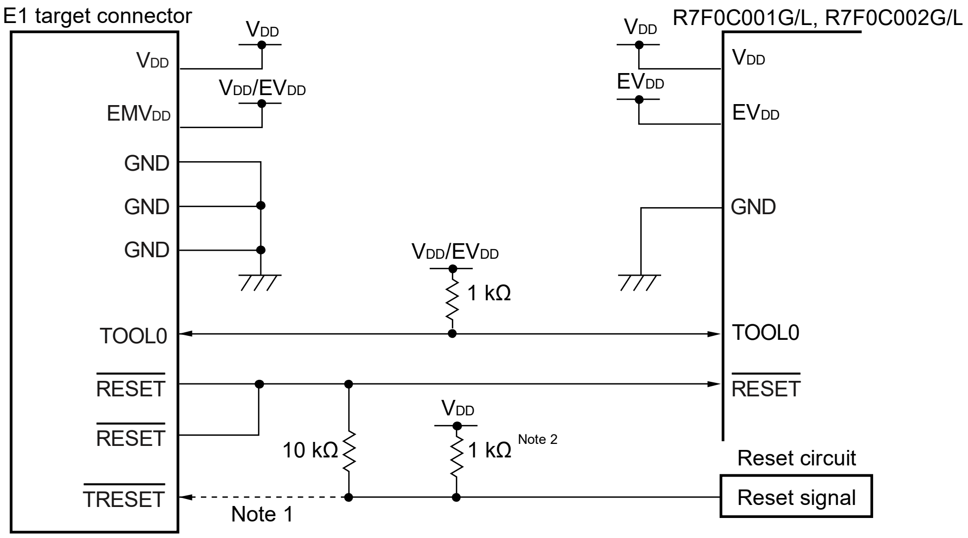
CHAPTER 24 ON-CHIP DEBUG FUNCTION

24.1 Connecting E1 On-chip Debugging Emulator to R7F0C001G/L, R7F0C002G/L

The R7F0C001G/L, R7F0C002G/L use the V_{DD}, $\overline{\text{RESET}}$, 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 R7F0C001G/L, R7F0C002G/L have 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 24-1. Connection Example of E1 On-chip Debugging Emulator and R7F0C001G/L, R7F0C002G/L



Notes 1. Connecting the dotted line is not necessary during flash programming.

2. If the reset circuit on the target system does not have a buffer and generates a reset signal only with resistors and capacitors, this pull-up resistor is not necessary.

Caution **This circuit diagram is assumed that the reset signal outputs from an N-ch O.D. buffer (output resistor: 100 Ω or less)**

24.2 On-Chip Debug Security ID

The R7F0C001G/L, R7F0C002G/L have an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 22 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

Table 24-1. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes (except All FFH)

24.3 Securing of User Resources

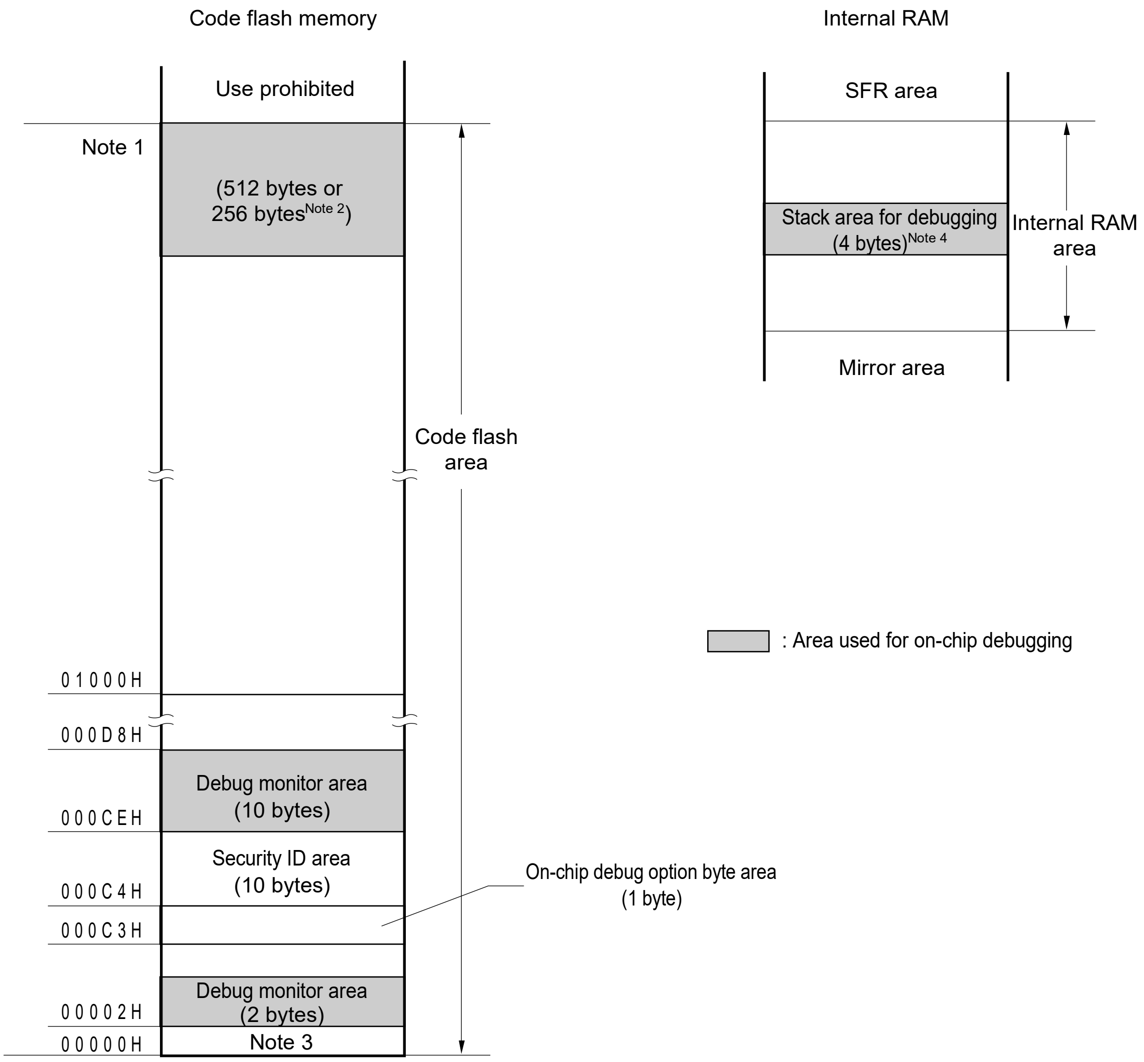
To perform communication between the R7F0C001G/L, R7F0C002G/L and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using linker options.

(1) Securement of memory space

The shaded portions in Figure 24-2 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 24-2. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on products as follows.

Products (Code Flash Memory Capacity)	Address of **Note 1**
R7F0C001G, R7F0C001L	03FFFFH
R7F0C002G, R7F0C002L	07FFFFH

2. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

3. In debugging, reset vector is rewritten to address allocated to a monitor program.

4. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 4 extra bytes are consumed for the stack area used. When using self-programming, 12 extra bytes are consumed for the stack area used.

CHAPTER 25 BCD CORRECTION CIRCUIT

25.1 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).

25.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

25.2.1 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 25-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

25.3 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 26 INSTRUCTION SET

This chapter lists the instructions in the R7F0C001G/L, R7F0C002G/L instruction set. For details of each operation and operation code, refer to the separate document **RL78 Family User's Manual: software (R01US0015)**.

26.1 Conventions Used in Operation List

26.1.1 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 26-1. Operand Identifiers and Specification Methods

Identifier	Description Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol (SFR symbol) FFF00H to FFFFFH Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only[Note]) FFF00H to FFFFFH
saddr saddrp	FFE20H to FFF1FH Immediate data or labels FFE20H to FF1FH Immediate data or labels (even addresses only[Note])
addr20 addr16 addr5	00000H to FFFFFH Immediate data or labels 0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions[Note]) 0080H to 00BFH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

Remark The special function registers can be described to operand sfr as symbols. See **Table 3-5 SFR List** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Table 3-6 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

26.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 26-2. 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)

26.1.3 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 26-3. 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

26.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

A interrupt and DMA transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 26-4. 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.

26.2 Operation List

Table 26-5. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	–	r ← byte			
		PSW, #byte	3	3	–	PSW ← byte	×	×	×
		CS, #byte	3	1	–	CS ← byte			
		ES, #byte	2	1	–	ES ← byte			
		!addr16, #byte	4	1	–	(addr16) ← byte			
		ES:!addr16, #byte	5	2	–	(ES, addr16) ← byte			
		saddr, #byte	3	1	–	(saddr) ← byte			
		sfr, #byte	3	1	–	sfr ← byte			
		[DE+byte], #byte	3	1	–	(DE+byte) ← byte			
		ES:[DE+byte],#byte	4	2	–	((ES, DE)+byte) ← byte			
		[HL+byte], #byte	3	1	–	(HL+byte) ← byte			
		ES:[HL+byte],#byte	4	2	–	((ES, HL)+byte) ← byte			
		[SP+byte], #byte	3	1	–	(SP+byte) ← byte			
		word[B], #byte	4	1	–	(B+word) ← byte			
		ES:word[B], #byte	5	2	–	((ES, B)+word) ← byte			
		word[C], #byte	4	1	–	(C+word) ← byte			
		ES:word[C], #byte	5	2	–	((ES, C)+word) ← byte			
		word[BC], #byte	4	1	–	(BC+word) ← byte			
		ES:word[BC], #byte	5	2	–	((ES, BC)+word) ← byte			
		A, r [Note 3]	1	1	–	A ← r			
		r, A [Note 3]	1	1	–	r ← A			
		A, PSW	2	1	–	A ← PSW			
		PSW, A	2	3	–	PSW ← A	×	×	×
		A, CS	2	1	–	A ← CS			
		CS, A	2	1	–	CS ← A			
		A, ES	2	1	–	A ← ES			
		ES, A	2	1	–	ES ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		A, ES:!addr16	4	2	5	A ← (ES, addr16)			
		!addr16, A	3	1	–	(addr16) ← A			
		ES:!addr16, A	4	2	–	(ES, addr16) ← A			
		A, saddr	2	1	–	A ← (saddr)			
		saddr, A	2	1	–	(saddr) ← A			

Notes 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

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.

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 26-5. Operation List (2/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit data transfer	MOV	A, sfr	2	1	—	A ← sfr			
		sfr, A	2	1	—	sfr ← A			
		A, [DE]	1	1	4	A ← (DE)			
		[DE], A	1	1	—	(DE) ← A			
		A, ES:[DE]	2	2	5	A ← (ES, DE)			
		ES:[DE], A	2	2	—	(ES, DE) ← A			
		A, [HL]	1	1	4	A ← (HL)			
		[HL], A	1	1	—	(HL) ← A			
		A, ES:[HL]	2	2	5	A ← (ES, HL)			
		ES:[HL], A	2	2	—	(ES, HL) ← A			
		A, [DE+byte]	2	1	4	A ← (DE + byte)			
		[DE+byte], A	2	1	—	(DE + byte) ← A			
		A, ES:[DE+byte]	3	2	5	A ← ((ES, DE) + byte)			
		ES:[DE+byte], A	3	2	—	((ES, DE) + byte) ← A			
		A, [HL+byte]	2	1	4	A ← (HL + byte)			
		[HL+byte], A	2	1	—	(HL + byte) ← A			
		A, ES:[HL+byte]	3	2	5	A ← ((ES, HL) + byte)			
		ES:[HL+byte], A	3	2	—	((ES, HL) + byte) ← A			
		A, [SP+byte]	2	1	—	A ← (SP + byte)			
		[SP+byte], A	2	1	—	(SP + byte) ← A			
		A, word[B]	3	1	4	A ← (B + word)			
		word[B], A	3	1	—	(B + word) ← A			
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)			
		ES:word[B], A	4	2	—	((ES, B) + word) ← A			
		A, word[C]	3	1	4	A ← (C + word)			
		word[C], A	3	1	—	(C + word) ← A			
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)			
		ES:word[C], A	4	2	—	((ES, C) + word) ← A			
		A, word[BC]	3	1	4	A ← (BC + word)			
		word[BC], A	3	1	—	(BC + word) ← A			
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)			
		ES:word[BC], A	4	2	—	((ES, BC) + word) ← A			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (3/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit data transfer	MOV	A, [HL+B]	2	1	4	A ← (HL + B)			
		[HL+B], A	2	1	–	(HL + B) ← A			
		A, ES:[HL+B]	3	2	5	A ← ((ES, HL) + B)			
		ES:[HL+B], A	3	2	–	((ES, HL) + B) ← A			
		A, [HL+C]	2	1	4	A ← (HL + C)			
		[HL+C], A	2	1	–	(HL + C) ← A			
		A, ES:[HL+C]	3	2	5	A ← ((ES, HL) + C)			
		ES:[HL+C], A	3	2	–	((ES, HL) + C) ← A			
		X, !addr16	3	1	4	X ← (addr16)			
		X, ES:!addr16	4	2	5	X ← (ES, addr16)			
		X, saddr	2	1	–	X ← (saddr)			
		B, !addr16	3	1	4	B ← (addr16)			
		B, ES:!addr16	4	2	5	B ← (ES, addr16)			
		B, saddr	2	1	–	B ← (saddr)			
		C, !addr16	3	1	4	C ← (addr16)			
		C, ES:!addr16	4	2	5	C ← (ES, addr16)			
		C, saddr	2	1	–	C ← (saddr)			
		ES, saddr	3	1	–	ES ← (saddr)			
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	–	A ←→ r			
		A, !addr16	4	2	–	A ←→ (addr16)			
		A, ES:!addr16	5	3	–	A ←→ (ES, addr16)			
		A, saddr	3	2	–	A ←→ (saddr)			
		A, sfr	3	2	–	A ←→ sfr			
		A, [DE]	2	2	–	A ←→ (DE)			
		A, ES:[DE]	3	3	–	A ←→ (ES, DE)			
		A, [HL]	2	2	–	A ←→ (HL)			
		A, ES:[HL]	3	3	–	A ←→ (ES, HL)			
		A, [DE+byte]	3	2	–	A ←→ (DE + byte)			
		A, ES:[DE+byte]	4	3	–	A ←→ ((ES, DE) + byte)			
		A, [HL+byte]	3	2	–	A ←→ (HL + byte)			
		A, ES:[HL+byte]	4	3	–	A ←→ ((ES, HL) + byte)			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (4/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit data transfer	XCH	A, [HL+B]	2	2	–	A ←→ (HL+B)			
		A, ES:[HL+B]	3	3	–	A ←→ ((ES, HL)+B)			
		A, [HL+C]	2	2	–	A ←→ (HL+C)			
		A, ES:[HL+C]	3	3	–	A ←→ ((ES, HL)+C)			
	ONEB	A	1	1	–	A ← 01H			
		X	1	1	–	X ← 01H			
		B	1	1	–	B ← 01H			
		C	1	1	–	C ← 01H			
		!addr16	3	1	–	(addr16) ← 01H			
		ES:!addr16	4	2	–	(ES, addr16) ← 01H			
		saddr	2	1	–	(saddr) ← 01H			
	CLRB	A	1	1	–	A ← 00H			
		X	1	1	–	X ← 00H			
		B	1	1	–	B ← 00H			
		C	1	1	–	C ← 00H			
		!addr16	3	1	–	(addr16) ← 00H			
		ES:!addr16	4	2	–	(ES,addr16) ← 00H			
		saddr	2	1	–	(saddr) ← 00H			
	MOVS	[HL+byte], X	3	1	–	(HL+byte) ← X	×		×
		ES:[HL+byte], X	4	2	–	(ES, HL+byte) ← X	×		×
16-bit data transfer	MOVW	rp, #word	3	1	–	rp ← word			
		saddrp, #word	4	1	–	(saddrp) ← word			
		sfrp, #word	4	1	–	sfrp ← word			
		AX, rp Note 3	1	1	–	AX ← rp			
		rp, AX Note 3	1	1	–	rp ← AX			
		AX, !addr16	3	1	4	AX ← (addr16)			
		!addr16, AX	3	1	–	(addr16) ← AX			
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)			
		ES:!addr16, AX	4	2	–	(ES, addr16) ← AX			
		AX, saddrp	2	1	–	AX ← (saddrp)			
		saddrp, AX	2	1	–	(saddrp) ← AX			
		AX, sfrp	2	1	–	AX ← sfrp			
		sfrp, AX	2	1	–	sfrp ← AX			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except rp = AX

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (5/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
16-bit data transfer	MOVW	AX, [DE]	1	1	4	AX ← (DE)			
		[DE], AX	1	1	–	(DE) ← AX			
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)			
		ES:[DE], AX	2	2	–	(ES, DE) ← AX			
		AX, [HL]	1	1	4	AX ← (HL)			
		[HL], AX	1	1	–	(HL) ← AX			
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)			
		ES:[HL], AX	2	2	–	(ES, HL) ← AX			
		AX, [DE+byte]	2	1	4	AX ← (DE+byte)			
		[DE+byte], AX	2	1	–	(DE+byte) ← AX			
		AX, ES:[DE+byte]	3	2	5	AX ← ((ES, DE) + byte)			
		ES:[DE+byte], AX	3	2	–	((ES, DE) + byte) ← AX			
		AX, [HL+byte]	2	1	4	AX ← (HL + byte)			
		[HL+byte], AX	2	1	–	(HL + byte) ← AX			
		AX, ES:[HL+byte]	3	2	5	AX ← ((ES, HL) + byte)			
		ES:[HL+byte], AX	3	2	–	((ES, HL) + byte) ← AX			
		AX, [SP+byte]	2	1	–	AX ← (SP + byte)			
		[SP+byte], AX	2	1	–	(SP + byte) ← AX			
		AX, word[B]	3	1	4	AX ← (B + word)			
		word[B], AX	3	1	–	(B+ word) ← AX			
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	–	((ES, B) + word) ← AX			
		AX, word[C]	3	1	4	AX ← (C + word)			
		word[C], AX	3	1	–	(C + word) ← AX			
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	–	((ES, C) + word) ← AX			
		AX, word[BC]	3	1	4	AX ← (BC + word)			
		word[BC], AX	3	1	–	(BC + word) ← AX			
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	–	((ES, BC) + word) ← AX			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (6/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
16-bit data transfer	MOVW	BC, !addr16	3	1	4	BC ← (addr16)			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
		BC, saddrp	2	1	–	BC ← (saddrp)			
		DE, saddrp	2	1	–	DE ← (saddrp)			
		HL, saddrp	2	1	–	HL ← (saddrp)			
	XCHW	AX, rp **Note 3**	1	1	–	AX ←→ rp			
	ONEW	AX	1	1	–	AX ← 0001H			
		BC	1	1	–	BC ← 0001H			
	CLRW	AX	1	1	–	AX ← 0000H			
		BC	1	1	–	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr)+byte	×	×	×
		A, r **Note 4**	2	1	–	A, CY ← A + r	×	×	×
		r, A	2	1	–	r, CY ← r + A	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A+ (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A + ((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A + (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A+((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A + (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A + ((ES, HL) + C)	×	×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except rp = AX

4. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (7/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit operation	ADDC	A, #byte	2	1	–	A, CY ← A+byte+CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) +byte+CY	×	×	×
		A, rv Note 3	2	1	–	A, CY ← A + r + CY	×	×	×
		r, A	2	1	–	r, CY ← r + A + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)+CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)+CY	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)+CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A+ (HL) + CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A+ (ES, HL) + CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A+ (HL+byte) + CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A+ ((ES, HL)+byte) + CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A+ (HL+B) +CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A+((ES, HL)+B)+CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A+ (HL+C)+CY	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A+ ((ES, HL)+C)+CY	×	×	×
	SUB	A, #byte	2	1	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r	×	×	×
		r, A	2	1	–	r, CY ← r – A	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES, addr16)	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A – (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A – (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A – ((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A – (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A – ((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A – (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A – ((ES, HL)+C)	×	×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (8/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	—	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	1	—	A, CY ← A – r – CY	×	×	×
		r, A	2	1	—	r, CY ← r – A – CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16) – CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES, addr16) – CY	×	×	×
		A, saddr	2	1	—	A, CY ← A – (saddr) – CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL) – CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A – (ES, HL) – CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A – (HL+byte) – CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A – ((ES, HL)+byte) – CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A – (HL+B) – CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A – ((ES, HL)+B) – CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A – (HL+C) – CY	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY ← A – ((ES:HL)+C) – CY	×	×	×
	AND	A, #byte	2	1	—	A ← A ∧ byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	—	A ← A ∧ r	×		
		r, A	2	1	—	R ← r ∧ A	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, saddr	2	1	—	A ← A ∧ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ∧ (HL+byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ∧ ((ES:HL)+byte)	×		
		A, [HL+B]	2	1	4	A ← A ∧ (HL+B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ∧ ((ES:HL)+B)	×		
		A, [HL+C]	2	1	4	A ← A ∧ (HL+C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∧ ((ES:HL)+C)	×		

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (9/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit operation	OR	A, #byte	2	1	—	A ← A∨byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr)∨byte	×		
		A, r Note 3	2	1	—	A ← A∨r	×		
		r, A	2	1	—	r ← r∨A	×		
		A, !addr16	3	1	4	A ← A∨(addr16)	×		
		A, ES:!addr16	4	2	5	A ← A∨(ES:addr16)	×		
		A, saddr	2	1	—	A ← A∨(saddr)	×		
		A, [HL]	1	1	4	A ← A∨(H)	×		
		A, ES:[HL]	2	2	5	A ← A∨(ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A∨(HL+byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A∨((ES:HL)+byte)	×		
		A, [HL+B]	2	1	4	A ← A∨(HL+B)	×		
		A, ES:[HL+B]	3	2	5	A ← A∨((ES:HL)+B)	×		
		A, [HL+C]	2	1	4	A ← A∨(HL+C)	×		
		A, ES:[HL+C]	3	2	5	A ← A∨((ES:HL)+C)	×		
	XOR	A, #byte	2	1	—	A ← A⊻byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr)⊻byte	×		
		A, r Note 3	2	1	—	A ← A⊻r	×		
		r, A	2	1	—	r ← r⊻A	×		
		A, !addr16	3	1	4	A ← A⊻(addr16)	×		
		A, ES:!addr16	4	2	5	A ← A⊻(ES:addr16)	×		
		A, saddr	2	1	—	A ← A⊻(saddr)	×		
		A, [HL]	1	1	4	A ← A⊻(HL)	×		
		A, ES:[HL]	2	2	5	A ← A⊻(ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A⊻(HL+byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A⊻((ES:HL)+byte)	×		
		A, [HL+B]	2	1	4	A ← A⊻(HL+B)	×		
		A, ES:[HL+B]	3	2	5	A ← A⊻((ES:HL)+B)	×		
		A, [HL+C]	2	1	4	A ← A⊻(HL+C)	×		
		A, ES:[HL+C]	3	2	5	A ← A⊻((ES:HL)+C)	×		

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (10/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	–	A – byte	×	×	×
		!addr16, #byte	4	1	4	(addr16) – byte	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	×	×	×
		saddr, #byte	3	1	–	(saddr) – byte	×	×	×
		A, r Note3	2	1	–	A – r	×	×	×
		r, A	2	1	–	r – A	×	×	×
		A, !addr16	3	1	4	A – (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A – (ES:addr16)	×	×	×
		A, saddr	2	1	–	A – (saddr)	×	×	×
		A, [HL]	1	1	4	A – (HL)	×	×	×
		A, ES:[HL]	2	2	5	A – (ES:HL)	×	×	×
		A, [HL+byte]	2	1	4	A – (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A – ((ES:HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A – (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A – ((ES:HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A – (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A – ((ES:HL)+C)	×	×	×
	CMP0	A	1	1	–	A – 00H	×	0	0
		X	1	1	–	X – 00H	×	0	0
		B	1	1	–	B – 00H	×	0	0
		C	1	1	–	C – 00H	×	0	0
		!addr16	3	1	4	(addr16) – 00H	×	0	0
		ES:!addr16	4	2	5	(ES:addr16) – 00H	×	0	0
		saddr	2	1	–	(saddr) – 00H	×	0	0
	CMPS	X, [HL+byte]	3	1	4	X – (HL+byte)	×	×	×
		X, ES:[HL+byte]	4	2	5	X – ((ES:HL)+byte)	×	×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (11/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	1	–	AX, CY ← AX+word	×	×	×
		AX, AX	1	1	–	AX, CY ← AX+AX	×	×	×
		AX, BC	1	1	–	AX, CY ← AX+BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX+DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX+HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX+(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX+(ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX+(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX+(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX+((ES:HL)+byte)	×	×	×
	SUBW	AX, #word	3	1	–	AX, CY ← AX – word	×	×	×
		AX, BC	1	1	–	AX, CY ← AX – BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX – DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX – HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX – (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX – ((ES:HL)+byte)	×	×	×
	CMPW	AX, #word	3	1	–	AX – word	×	×	×
		AX, BC	1	1	–	AX – BC	×	×	×
		AX, DE	1	1	–	AX – DE	×	×	×
		AX, HL	1	1	–	AX – HL	×	×	×
		AX, !addr16	3	1	4	AX – (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX – (ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX – ((ES:HL)+byte)	×	×	×
Multiply	MULU	X	1	1	–	AX ← A×X			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (12/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
Increment/ decrement	INC	r	1	1	–	r ← r+1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16)+1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16)+1	×	×	
		saddr	2	2	–	(saddr) ← (saddr)+1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte)+1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte)+1	×	×	
	DEC	r	1	1	–	r ← r – 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) – 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) – 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte) – 1	×	×	
	INCW	rp	1	1	–	rp ← rp+1			
		!addr16	3	2	–	(addr16) ← (addr16)+1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16)+1			
		saddrp	2	2	–	(saddrp) ← (saddrp)+1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte)+1			
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte)+1			
	DECW	rp	1	1	–	rp ← rp – 1			
		!addr16	3	2	–	(addr16) ← (addr16) – 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) – 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1			
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte) – 1			
Shift	SHR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) ×cnt			×
	SHRW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) ×cnt			×
	SHL	A, cnt	2	1	–	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) ×cnt			×
		B, cnt	2	1	–	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) ×cnt			×
		C, cnt	2	1	–	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) ×cnt			×
	SHLW	AX, cnt	2	1	–	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) ×cnt			×
		BC, cnt	2	1	–	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) ×cnt			×
	SAR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) ×cnt			×
	SARW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← AX_{15}) ×cnt			×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remarks 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

2. cnt indicates the bit shift count.

Table 26-5. Operation List (13/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Rotate	ROR	A, 1	2	1	–	(CY, A_7 ← A_0, A_{m-1} ← A_m)×1			×
	ROL	A, 1	2	1	–	(CY, A_0 ← A_7, A_{m+1} ← A_m)×1			×
	RORC	A, 1	2	1	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m)×1			×
	ROLC	A, 1	2	1	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m)×1			×
	ROLWC	AX,1	2	1	–	(CY ← AX_{15}, AX_0 ← CY, AX_{m+1} ← AX_m) ×1			×
		BC,1	2	1	–	(CY ← BC_{15}, BC_0 ← CY, BC_{m+1} ← BC_m) ×1			×
Bit manipulate	MOV1	CY, A.bit	2	1	–	CY ← A.bit			×
		A.bit, CY	2	1	–	A.bit ← CY			
		CY, PSW.bit	3	1	–	CY ← PSW.bit			×
		PSW.bit, CY	3	4	–	PSW.bit ← CY	×	×	
		CY, saddr.bit	3	1	–	CY ← (saddr).bit			×
		saddr.bit, CY	3	2	–	(saddr).bit ← CY			
		CY, sfr.bit	3	1	–	CY ← sfr.bit			×
		sfr.bit, CY	3	2	–	sfr.bit ← CY			
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		[HL].bit, CY	2	2	–	(HL).bit ← CY			
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit			×
		ES:[HL].bit, CY	3	3	–	(ES, HL).bit ← CY			
	AND1	CY, A.bit	2	1	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∧ PSW.bit			×
		CY, saddr.bit	3	1	–	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∧ sfr.bit			×
		CY,[HL].bit	2	1	4	CY ← CY ∧ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit			×
	OR1	CY, A.bit	2	1	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∨ PSW.bit			×
		CY, saddr.bit	3	1	–	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∨ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (14/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
Bit manipulate	XOR1	CY, A.bit	2	1	–	CY ← CY ⊻ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ⊻ PSW.bit			×
		CY, saddr.bit	3	1	–	CY ← CY ⊻ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ⊻ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ⊻ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ⊻ (ES, HL).bit			×
	SET1	A.bit	2	1	–	A.bit ← 1			
		PSW.bit	3	4	–	PSW.bit ← 1	×	×	×
		!addr16.bit	4	2	–	(addr16).bit ← 1			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 1			
		saddr.bit	3	2	–	(saddr).bit ← 1			
		sfr.bit	3	2	–	sfr.bit ← 1			
		[HL].bit	2	2	–	(HL).bit ← 1			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 1			
	CLR1	A.bit	2	1	–	A.bit ← 0			
		PSW.bit	3	4	–	PSW.bit ← 0	×	×	×
		!addr16.bit	4	2	–	(addr16).bit ← 0			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 0			
		saddr.bit	3	2	–	(saddr.bit) ← 0			
		sfr.bit	3	2	–	sfr.bit ← 0			
		[HL].bit	2	2	–	(HL).bit ← 0			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 0			
	SET1	CY	2	1	–	CY ← 1			1
	CLR1	CY	2	1	–	CY ← 0			0
	NOT1	CY	2	1	–	CY ← $\overline{\text{CY}}$			×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (15/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
Call/ return	CALL	rp	2	3	–	(SP – 2) ← $(PC+2)_S$, (SP – 3) ← $(PC+2)_H$, (SP – 4) ← $(PC+2)_L$, PC ← CS, rp, SP ← SP – 4			
		$!addr20	3	3	–	(SP – 2) ← $(PC+3)_S$, (SP – 3) ← $(PC+3)_H$, (SP – 4) ← $(PC+3)_L$, PC ← PC+3+jdisp16, SP ← SP – 4			
		!addr16	3	3	–	(SP – 2) ← $(PC+3)_S$, (SP – 3) ← $(PC+3)_H$, (SP – 4) ← $(PC+3)_L$, PC ← 0000, addr16, SP ← SP – 4			
		!!addr20	4	3	–	(SP – 2) ← $(PC+4)_S$, (SP – 3) ← $(PC+4)_H$, (SP – 4) ← $(PC+4)_L$, PC ← addr20, SP ← SP – 4			
	CALLT	[addr5]	2	5	–	(SP – 2) ← $(PC+2)_S$, (SP – 3) ← $(PC+2)_H$, (SP – 4) ← $(PC+2)_L$, PC_S ← 0000, PC_H ← (0000, addr5+1), PC_L ← (0000, addr5), SP ← SP – 4			
	BRK	-	2	5	–	(SP – 1) ← PSW, (SP – 2) ← $(PC+2)_S$, (SP – 3) ← $(PC+2)_H$, (SP – 4) ← $(PC+2)_L$, PC_S ← 0000, PC_H ← (0007FH), PC_L ← (0007EH), SP ← SP – 4, IE ← 0			
	RET	-	1	6	–	PC_L ← (SP), PC_H ← (SP+1), PC_S ← (SP+2), SP ← SP+4			
	RETI	-	2	6	–	PC_L ← (SP), PC_H ← (SP+1), PC_S ← (SP+2), PSW ← (SP+3), SP ← SP+4	R	R	R
	RETB	-	2	6	–	PC_L ← (SP), PC_H ← (SP+1), PC_S ← (SP+2), PSW ← (SP+3), SP ← SP+4	R	R	R

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulate	PUSH	PSW	2	1	–	(SP – 1) ← PSW, (SP – 2) ← 00H, SP ← SP – 2			
		rp	1	1	–	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2			
	POP	PSW	2	3	–	PSW ← (SP+1), SP ← SP + 2	R	R	R
		rp	1	1	–	rp_L ←(SP), rp_H ← (SP+1), SP ← SP + 2			
	MOVW	SP, #word	4	1	–	SP ← word			
		SP, AX	2	1	–	SP ← AX			
		AX, SP	2	1	–	AX ← SP			
		HL, SP	3	1	–	HL ← SP			
		BC, SP	3	1	–	BC ← SP			
		DE, SP	3	1	–	DE ← SP			
	ADDW	SP, #byte	2	1	–	SP ← SP + byte			
	SUBW	SP, #byte	2	1	–	SP ← SP – byte			
Un-conditional branch	BR	AX	2	3	–	PC ← CS, AX			
		$addr20	2	3	–	PC ← PC + 2 + jdisp8			
		$!addr20	3	3	–	PC ← PC + 3 + jdisp16			
		!addr16	3	3	–	PC ← 0000, addr16			
		!!addr20	4	3	–	PC ← addr20			
Conditional branch	BC	$addr20	2	2/4 [Note3]	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4 [Note3]	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4 [Note3]	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4 [Note3]	–	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4 [Note3]	–	PC ← PC + 3 + jdisp8 if (Z∨CY)=0			
	BNH	$addr20	3	2/4 [Note3]	–	PC ← PC + 3 + jdisp8 if (Z∨CY)=1			
	BT	saddr.bit, $addr20	4	3/5 [Note3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5 [Note3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5 [Note3]	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5 [Note3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5 [Note3]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6 [Note3]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 26-5. Operation List (17/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5 Note3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5 Note3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5 Note3	—	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5 Note3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5 Note3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6 Note3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5 Note3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5 Note3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5 Note3	—	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5 Note3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5 Note3	—	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6 Note3	—	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit			
Conditional skip	SKC	—	2	1	—	Next instruction skip if CY = 1			
	SKNC	—	2	1	—	Next instruction skip if CY = 0			
	SKZ	—	2	1	—	Next instruction skip if Z = 1			
	SKNZ	—	2	1	—	Next instruction skip if Z = 0			
	SKH	—	2	1	—	Next instruction skip if (Z∨CY)=0			
	SKNH	—	2	1	—	Next instruction skip if (Z∨CY)=1			
CPU control	SEL Note4	RBn	2	1	—	RBS[1:0] ← n			
	NOP	—	1	1	—	No Operation			
	EI	—	3	4	—	IE ← 1 (Enable Interrupt)			
	DI	—	3	4	—	IE ← 0 (Disable Interrupt)			
	HALT	—	2	3	—	Set HALT Mode			
	STOP	—	2	3	—	Set STOP Mode			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

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 27 ELECTRICAL SPECIFICATIONS

Cautions 1. The R7F0C001G/L, R7F0C002G/L have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. With products not provided with an EV_{DD}, or EV_{SS} pin, replace EV_{DD} with V_{DD}, or replace EV_{SS} with V_{SS}.

3. The pins mounted depend on the product. Refer to 2.1 Port Function to 2.2.1 Pins for each product (pins other than port pins).

27.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) **(1/3)**

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}	$V_{DD} = EV_{DD}$	–0.5 to +6.5	V
	EV_{DD}	$V_{DD} = EV_{DD}$	–0.5 to +6.5	V
	EV_{SS}		–0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC	–0.3 to +2.8 and –0.3 to V_{DD} + 0.3$^{\text{Note 1}}$	V
Input voltage	V_{I1}	P10 to P17, P30 to P32, P40 to P43, P50 to P54, P70 to P74, P120, P125 to P127,P140 to P147	–0.3 to EV_{DD} +0.3 and –0.3 to V_{DD} + 0.3$^{\text{Note 2}}$	V
	V_{I2}	P60, P61 (N-ch open-drain)	–0.3 to EV_{DD} +0.3 and –0.3 to V_{DD} + 0.3$^{\text{Note 2}}$	V
	V_{I3}	P20, P21, P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$	–0.3 to V_{DD} + 0.3$^{\text{Note 2}}$	V
Output voltage	V_{O1}	P10 to P17, P30 to P32, P40 to P43, P50 to P54, P60, P61, P70 to P74, P120, P125 to P127, P130, P140 to P147	–0.3 to EV_{DD} + 0.3 and –0.3 to V_{DD} + 0.3$^{\text{Note 2}}$	V
	V_{O2}	P20, P21	–0.3 to V_{DD} + 0.3$^{\text{Note 2}}$	V
Analog input voltage	V_{AI1}	ANI16 to ANI23	–0.3 to EV_{DD} + 0.3 and –0.3 to $AV_{REF(+)}$ + 0.3 Notes 2, 3	V
	V_{AI2}	ANI0, ANI1	–0.3 to V_{DD} + 0.3 and –0.3 to $AV_{REF(+)}$ + 0.3 Notes 2, 3	V

Notes 1. Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

2. Must be 6.5 V or lower.

3. Do not exceed $AV_{REF(+)}$ + 0.3 V in case of A/D conversion target pin.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

2. $AV_{REF(+)}$: + side reference voltage of the A/D converter.

3. V_{SS} : Reference voltage

Absolute Maximum Ratings (T_A = 25°C) **(2/3)**

Parameter	Symbols	Conditions		Ratings	Unit
LCD voltage	V_{L1}	V_{L1} voltage$^{\text{Note 1}}$		–0.3 to +2.8 and –0.3 to V_{L4} + 0.3	V
	V_{L2}	V_{L2} voltage$^{\text{Note 1}}$		–0.3 to V_{L4} + 0.3 $^{\text{Note 2}}$	V
	V_{L3}	V_{L3} voltage$^{\text{Note 1}}$		–0.3 to V_{L4} + 0.3 $^{\text{Note 2}}$	V
	V_{L4}	V_{L4} voltage$^{\text{Note 1}}$		–0.3 to +6.5	V
	V_{LCAP}	CAPL, CAPH voltage$^{\text{Note 1}}$		–0.3 to V_{L4} + 0.3 $^{\text{Note 2}}$	V
	V_{LOUT}	COM0 to COM7, SEG0 to SEG38, output voltage	External resistance division method	–0.3 to V_{DD} + 0.3 $^{\text{Note 2}}$	V
			Capacitor split method	–0.3 to V_{DD} + 0.3 $^{\text{Note 2}}$	
			Internal voltage boosting method	–0.3 to V_{L4} + 0.3 $^{\text{Note 2}}$	

Notes **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 μF ± 30%) and connect a capacitor (0.47 μF ± 30%) between the CAPL and CAPH pins.

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.**

Remark V_{SS} : Reference voltage

Absolute Maximum Ratings (T_A = 25°C) (3/3)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P10 to P17, P30 to P32, P40 to P43, P50 to P54, P70 to P74, P120, P125 to P127, P130, P140 to P147	–40	mA
		Total of all pins –170 mA	P10 to P14, P40 to P43, P120, P130, P140 to P147	–70	mA
			P15 to P17, P30 to P32, P50 to P54, P70 to P74, P125 to P127	–100	mA
	I_{OH2}	Per pin	P20, P21	–0.5	mA
		Total of all pins		–1	mA
Output current, low	I_{OL1}	Per pin	P10 to P17, P30 to P32, P40 to P43, P50 to P54, P60, P61, P70 to P74, P120, P125 to P127, P130, P140 to P147	40	mA
		Total of all pins 170 mA	P10 to P14, P40 to P43, P120, P130, P140 to P147	70	mA
			P15 to P17, P30 to P32, P50 to P54, P60, P61, P70 to P74, P125 to P127	100	mA
	I_{OL2}	Per pin	P20, P21	1	mA
		Total of all pins		2	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.

27.2 Oscillator Characteristics

27.2.1 X1, XT1 oscillator characteristics

(T_A = –40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X)[Note]	Ceramic resonator/ crystal resonator	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20.0	MHz
		2.4 V ≤ V_{DD} ≤ 2.7 V	1.0		16.0	MHz
		1.8 V ≤ V_{DD} < 2.7 V	1.0		8.0	MHz
		1.6 V ≤ V_{DD} <1.8 V	1.0		4.0	MHz
XT1 clock oscillation frequency (f_{XT})[Note]	Crystal resonator		32	32.768	35	kHz

Note Indicates only permissible oscillator frequency ranges. Refer to **27.4 AC Characteristics** for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark When using the X1 oscillator and XT1 oscillator, refer to **5.4 System Clock Oscillator**.

27.2.2 On-chip oscillator characteristics

(T_A = –40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

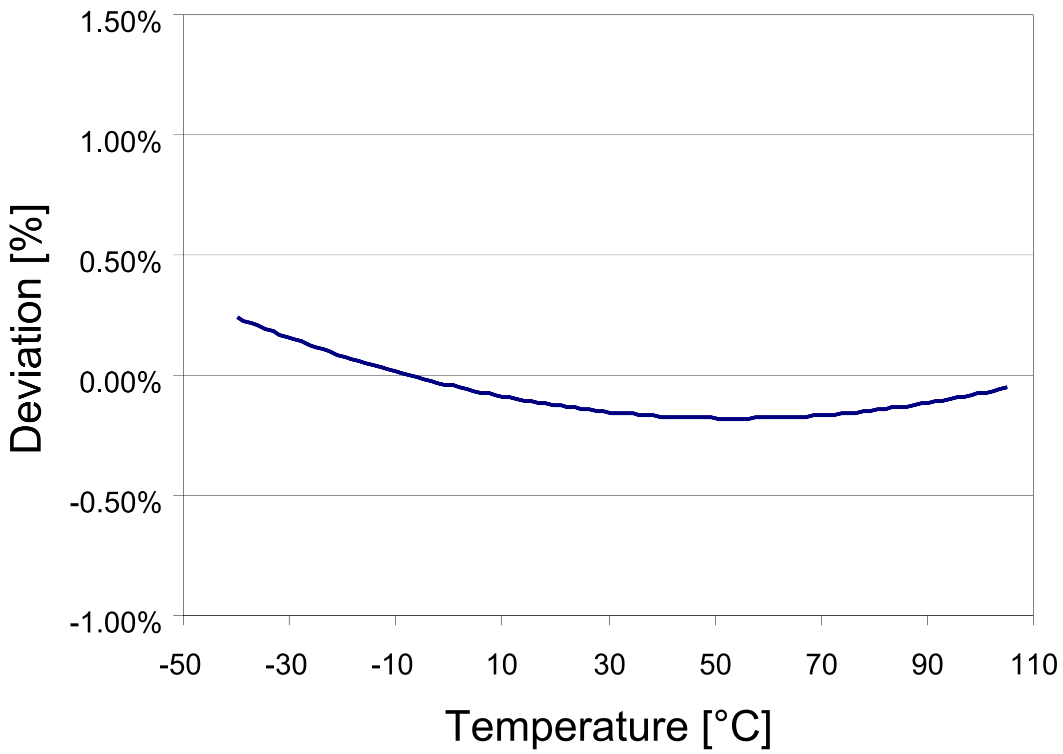
Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency [Note 1]	f_{IH}			1		24	MHz
High-speed on-chip oscillator clock frequency accuracy [Note 2]		–20 to +85°C	1.8 V ≤ V_{DD} ≤ 5.5 V	–1	±0.5 [Note 3]	+1	%
			1.6 V ≤ V_{DD} < 1.8 V	–5		+5	%
		–40 to –20°C	1.8 V ≤ V_{DD} ≤ 5.5 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	%

Notes 1. High-speed on-chip oscillator frequency is selected by bits 0 to 3 of option byte (000C2H) and bits 0 to 2 of HOCODIV register.

2. This indicates the oscillator characteristics only. Refer to **27.4 AC Characteristics** for instruction execution time.

3. T_A = +25°C, V_{DD} = 3.0 V, and 24 MHz selected.

Deviation of high-speed on-chip oscillator clock frequency



Caution The above mentioned value is only for your reference. The value was measured under certain conditions and does not guarantee the product's characteristics.

27.3 DC Characteristics

27.3.1 Pin characteristics

(T_A = –40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) (1/5)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P10 to P17, P30 to P32, P40 to P43, P50 to P54, P70 to P74, P120, P125 to P127, P130, P140 to P147					–10.0 Note 2	mA
		Total of P10 to P14, P40 to P43, P120, P130, P140 to P147 (When duty ≤ 70% [Note 3])		4.0 V ≤ EV_{DD} ≤ 5.5 V			–40.0	mA
				2.7 V ≤ EV_{DD} < 4.0 V			–8.0	mA
				1.8 V ≤ EV_{DD} < 2.7 V			–4.0	mA
				1.6 V ≤ EV_{DD} < 1.8 V			–2.0	mA
		Total of P15 to P17, P30 to P32, P50 to P54, P70 to P74, P125 to P127 (When duty ≤ 70% [Note 3])		4.0 V ≤ EV_{DD} ≤ 5.5 V			–60.0	mA
				2.7 V ≤ EV_{DD} < 4.0 V			–15.0	mA
				1.8 V ≤ EV_{DD} < 2.7 V			–8.0	mA
				1.6 V ≤ EV_{DD} < 1.8 V			–4.0	mA
		Total of all pins (When duty ≤ 70%[Note 3])					–100.0	mA
	I_{OH2}	P20, P21	Per pin				–0.1	mA
			Total of all pins	1.6 V ≤ V_{DD} ≤ 5.5 V			–0.2	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from the V_{DD} and EV_{DD} pins to an output pin.

2. Do not exceed the total current value.

3. Specification under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OH} = –40.0 mA

Total output current of pins = (–40.0 × 0.7)/(80 × 0.01) ≈ –35.0 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 P10, P12, P15, and P17 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 ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) **(2/5)**

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P10 to P17, P30 to P32, P40 to P43, P50 to P54, P70 to P74, P120, P125 to P127, P130, P140 to P147					20.0 [Note 2]	mA
		Per pin for P60, P61					15.0 [Note 2]	mA
		Total of P10 to P14, P40 to P43, P120, P130, P140 to P147 (When duty ≤ 70% [Note 3])		4.0 V ≤ EV_{DD} ≤ 5.5 V			70.0	mA
				2.7 V ≤ EV_{DD} < 4.0 V			15.0	mA
				1.8 V ≤ EV_{DD} < 2.7 V			9.0	mA
				1.6 V ≤ EV_{DD} < 1.8 V			4.5	mA
		Total of P15 to P17, P30 to P32, P50 to P54, P60, P61, P70 to P74, P125 to P127 (When duty ≤ 70% [Note 3])		4.0 V ≤ EV_{DD} ≤ 5.5 V			80.0	mA
				2.7 V ≤ EV_{DD} < 4.0 V			35.0	mA
				1.8 V ≤ EV_{DD} < 2.7 V			20.0	mA
				1.6 V ≤ EV_{DD} < 1.8 V			10.0	mA
		Total of all pins (When duty ≤ 70% [Note 3])					150.0	mA
	I_{OL2}	P20, P21	Per pin				0.4	mA
			Total of all pins	1.6 V ≤ V_{DD} ≤ 5.5 V			0.8	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the EV_{SS} and V_{SS} pin.

2. Do not exceed the total current value.

3. Specification under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

<Example> Where n = 80% and I_{OL} = 70.0 mA

Total output current of pins = (70.0 × 0.7)/(80 × 0.01) ≈ 61.25 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 ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) **(3/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P10 to P17, P30 to P32, P40 to P43, P50 to P54, P70 to P74, P120, P125 to P127, P140 to P147	Normal input buffer	$0.8EV_{DD}$		EV_{DD}	V
	V_{IH2}	P10, P11, P15, P16	TTL input buffer 4.0 V ≤ EV_{DD} ≤ 5.5 V	2.2		EV_{DD}	V
			TTL input buffer 3.3 V ≤ EV_{DD} < 4.0 V	2.0		EV_{DD}	V
			TTL input buffer 1.6 V ≤ EV_{DD} < 3.3 V	1.50		EV_{DD}	V
	V_{IH3}	P20, P21		$0.7V_{DD}$		V_{DD}	V
	V_{IH4}	P60, P61		$0.7EV_{DD}$		EV_{DD}	V
	V_{IH5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		$0.8V_{DD}$		V_{DD}	V
Input voltage, low	V_{IL1}	P10 to P17, P30 to P32, P40 to P43, P50 to P54, P70 to P74, P120, P125 to P127, P140 to P147	Normal input buffer	0		$0.2EV_{DD}$	V
	V_{IL2}	P10, P11, P15, P16	TTL input buffer 4.0 V ≤ EV_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 3.3 V ≤ EV_{DD} < 4.0 V	0		0.5	V
			TTL input buffer 1.6 V ≤ EV_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P20, P21		0		$0.3V_{DD}$	V
	V_{IL4}	P60, P61		0		$0.3EV_{DD}$	V
	V_{IL5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0		$0.2V_{DD}$	V

Caution The maximum value of V_{IH} of P10, P12, P15, P17 is EV_{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 ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) **(4/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P10 to P17, P30 to P32, P40 to P43, P50 to P54, P70 to P74, P120, P125 to P127, P130, P140 to P147	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –10 mA	EV_{DD}–1.5			V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –3.0 mA	EV_{DD}–0.7			V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –2.0 mA	EV_{DD}–0.6			V
			1.8 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –1.5 mA	EV_{DD}–0.5			V
			1.6 V ≤ EV_{DD} ≤ 5.5 V, I_{OH1} = –1.0 mA	EV_{DD}–0.5			V
	V_{OH2}	P20, P21	1.6 V ≤ V_{DD} ≤ 5.5 V, I_{OH2} = –100 μA	V_{DD}–0.5			V
Output voltage, low	V_{OL1}	P10 to P17, P30 to P32, P40 to P43, P50 to P54, P70 to P74, P120, P125 to P127, P130, P140 to P147	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 20 mA			1.3	V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.6	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA			0.4	V
			1.8 V ≤ EV_{DD} ≤ 5.5 V, I_{OL1} = 0.6 mA			0.4	V
			1.6 V ≤ EV_{DD} < 5.5 V, I_{OL1} = 0.3 mA			0.4	V
	V_{OL2}	P20, P21	1.6 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 400 μA			0.4	V
	V_{OL3}	P60, P61	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 15.0 mA			2.0	V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 5.0 mA			0.4	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 3.0 mA			0.4	V
			1.8 V ≤ EV_{DD} ≤ 5.5 V, I_{OL3} = 2.0 mA			0.4	V
			1.6 V ≤ EV_{DD} < 5.5 V, I_{OL3} = 1.0 mA			0.4	V

Caution **P10, P12, P15, P17 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 ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) (5/5)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P10 to P17, P30 to P32, P40 to P43, P50 to P54, P60, P61, P70 to P74, P120, P125 to P127, P140 to P147	$V_I = EV_{DD}$				1	μA
	I_{LIH2}	P20, P21, P137, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{DD}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P10 to P17, P30 to P32, P40 to P43, P50 to P54, P60, P61, P70 to P74, P120, P125 to P127, P140 to P147	$V_I = EV_{SS}$				–1	μA
	I_{LIL2}	P20, P21, P137, $\overline{\text{RESET}}$	$V_I = V_{SS}$				–1	μA
	I_{LIL3}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{SS}$	In input port or external clock input			–1	μA
				In resonator connection			–10	μA
On-chip pll-up resistance	R_{U1}	$V_I = EV_{SS}$	SEGxx port					
				2.4 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V	10	20	100	kΩ
				1.6 V ≤ EV_{DD} = V_{DD} < 2.4 V	10	30	100	kΩ
	R_{U2}		Ports other than above (Except for P60, P61, and P130)		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

27.3.2 Supply current characteristics

(T_A = –40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) (1/3)

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	HS (high-speed main) mode Note 5	f_{IH} = 24 MHz Note 3	Basic operation	V_{DD} = 5.0 V		1.5		mA
						V_{DD} = 3.0 V		1.5		mA
					Normal operation	V_{DD} = 5.0 V		3.3	5.0	mA
						V_{DD} = 3.0 V		3.3	5.0	mA
				f_{IH} = 16 MHz Note 3	Normal operation	V_{DD} = 5.0 V		2.5	3.7	mA
						V_{DD} = 3.0 V		2.5	3.7	mA
			LS (low-speed main) mode Note 5	f_{IH} = 8 MHz Note 3	Normal operation	V_{DD} = 3.0 V		1.2	1.8	mA
						V_{DD} = 2.0 V		1.2	1.8	mA
			LV (low-voltage main) mode Note 5	f_{IH} = 4 MHz Note 3	Normal operation	V_{DD} = 3.0 V		1.2	1.7	mA
						V_{DD} = 2.0 V		1.2	1.7	mA
			HS (high-speed main) mode Note 5	f_{MX} = 20 MHz Note 2, V_{DD} = 5.0 V	Normal operation	Square wave input		2.8	4.4	mA
						Resonator connection		3.0	4.6	mA
				f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		2.8	4.4	mA
						Resonator connection		3.0	4.6	mA
				f_{MX} = 10 MHz Note 2, V_{DD} = 5.0 V	Normal operation	Square wave input		1.8	2.6	mA
						Resonator connection		1.8	2.6	mA
				f_{MX} = 10 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		1.8	2.6	mA
						Resonator connection		1.8	2.6	mA
			LS (low-speed main) mode Note 5	f_{MX} = 8 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		1.1	1.7	mA
						Resonator connection		1.1	1.7	mA
				f_{MX} = 8 MHz Note 2, V_{DD} = 2.0 V	Normal operation	Square wave input		1.1	1.7	mA
						Resonator connection		1.1	1.7	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 4 T_A = –40°C	Normal operation	Square wave input		3.5	4.9	μA
						Resonator connection		3.6	5.0	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +25°C	Normal operation	Square wave input		3.6	4.9	μA
						Resonator connection		3.7	5.0	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +50°C	Normal operation	Square wave input		3.7	5.5	μA
						Resonator connection		3.8	5.6	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +70°C	Normal operation	Square wave input		3.8	6.3	μA
						Resonator connection		3.9	6.4	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +85°C	Normal operation	Square wave input		4.1	7.7	μA
						Resonator connection		4.2	7.8	μA

(**Notes** and **Remarks** are listed on the next page.)

<R>

Notes 1. Total current flowing into V_{DD} and EV_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD}, EV_{DD} or V_{SS}, EV_{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 the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the “TYP.” and “MAX.” columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the RTC.

2. When high-speed on-chip oscillator and subsystem clock are stopped.

3. When high-speed system clock and subsystem clock are stopped.

<R>

4. When high-speed on-chip oscillator and high-speed system clock are stopped. When AMPHS1 = 1 (Ultra-low power consumption oscillation).

5. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: $2.7\ V \leq V_{DD} \leq 5.5\ V$@1 MHz to 24 MHz
$2.4\ V \leq V_{DD} \leq 5.5\ V$@1 MHz to 16 MHz
LS (low-speed main) mode: $1.8\ V \leq V_{DD} \leq 5.5\ V$@1 MHz to 8 MHz
LV (low-voltage main) mode: $1.6\ V \leq V_{DD} \leq 5.5\ V$@1 MHz to 4 MHz

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

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

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

4. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C

(T_A = –40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) (2/3)

	Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
	Supply current Note 1	I_{DD2} Note 2	HALT mode	HS (high-speed main) mode Note 6	f_{IH} = 24 MHz Note 4	V_{DD} = 5.0 V		0.44	1.28	mA
<R>						V_{DD} = 3.0 V		0.44	1.28	mA
					f_{IH} = 16 MHz Note 4	V_{DD} = 5.0 V		0.40	1.00	mA
						V_{DD} = 3.0 V		0.40	1.00	mA
				LS (low-speed main) mode Note 6	f_{IH} = 8 MHz Note 4	V_{DD} = 3.0 V		260	530	μA
<R>						V_{DD} = 2.0 V		260	530	μA
				LV (low-voltage main) mode Note 6	f_{IH} = 4 MHz Note 4	V_{DD} = 3.0 V		420	640	μA
<R>						V_{DD} = 2.0 V		420	640	μA
				HS (high-speed main) mode Note 6	f_{MX} = 20 MHz Note 3, V_{DD} = 5.0 V	Square wave input		0.28	1.00	mA
<R>						Resonator connection		0.45	1.17	mA
					f_{MX} = 20 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.28	1.00	mA
						Resonator connection		0.45	1.17	mA
					f_{MX} = 10 MHz Note 3, V_{DD} = 5.0 V	Square wave input		0.19	0.60	mA
						Resonator connection		0.26	0.67	mA
					f_{MX} = 10 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.19	0.60	mA
						Resonator connection		0.26	0.67	mA
				LS (low-speed main) mode Note 6	f_{MX} = 8 MHz Note 3, V_{DD} = 3.0 V	Square wave input		95	330	μA
<R>						Resonator connection		145	380	μA
					f_{MX} = 8 MHz Note 3, V_{DD} = 2.0 V	Square wave input		95	330	μA
						Resonator connection		145	380	μA
				Subsystem clock operation	f_{SUB} = 32.768 kHz Note 5, T_A = –40°C	Square wave input		0.31	0.57	μA
						Resonator connection		0.50	0.76	μA
					f_{SUB} = 32.768 kHz Note 5, T_A = +25°C	Square wave input		0.37	0.57	μA
						Resonator connection		0.56	0.76	μA
					f_{SUB} = 32.768 kHz Note 5, T_A = +50°C	Square wave input		0.46	1.17	μA
						Resonator connection		0.65	1.36	μA
					f_{SUB} = 32.768 kHz Note 5, T_A = +70°C	Square wave input		0.57	1.97	μA
						Resonator connection		0.76	2.16	μA
					f_{SUB} = 32.768 kHz Note 5, T_A = +85°C	Square wave input		0.85	3.37	μA
						Resonator connection		1.04	3.56	μA
<R>		I_{DD3}	STOP mode Note 7	T_A = –40°C				0.17	0.50	μA
				T_A = +25°C				0.23	0.50	μA
				T_A = +50°C				0.32	1.10	μA
				T_A = +70°C				0.43	1.90	μA
				T_A = +85°C				0.71	3.30	μA

(**Notes** and **Remarks** are listed on the next page.)

Notes 1. Total current flowing into V_{DD} and EV_{DD}, including the input leakage current flowing when the level of the input
<R> pin is fixed to V_{DD}, EV_{DD} or V_{SS}, EV_{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 the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the “TYP.” and “MAX.” columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the RTC.

In the STOP mode, the currents in both the “TYP.” and “MAX.” columns do not include the operating currents of the peripheral modules.

2. During HALT instruction execution by flash memory.

3. When high-speed on-chip oscillator and subsystem clock are stopped.

4. When high-speed system clock and subsystem clock are stopped.

<R> **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).

<R> **6.** Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: $2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$@1 MHz to 24 MHz
$2.4\text{ V} \leq V_{DD} \leq 5.5\text{ V}$@1 MHz to 16 MHz
LS (low-speed main) mode: $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$@1 MHz to 8 MHz
LV (low-voltage main) mode: $1.6\text{ V} \leq V_{DD} \leq 5.5\text{ V}$@1 MHz to 4 MHz

<R> **7.** Regarding the value for current to operate the subsystem clock in STOP mode, refer to that in HALT mode.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

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

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

4. Except subsystem clock operation and STOP mode, temperature condition of the TYP. value is $T_A = 25°C$

(T_A = –40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) (3/3)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.20		μA
RTC operating current	I_{RTC} Notes 1, 2, 3	f_{MAIN} is stopped			0.08		μA
12-bit interval timer current	I_{IT} Notes 1, 2, 4				0.08		μA
Watchdog timer operating current	I_{WDT} Notes 1, 2, 5	f_{IL} = 15 kHz			0.24		μA
A/D converter operating current	I_{ADC} Notes 1, 6	When conversion at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V		1.3	1.7	mA
			Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		0.5	0.7	mA
A/D converter reference voltage current	I_{ADREF} Note 1				75.0		μA
Temperature sensor operating current	I_{TMPS} Note 1				75.0		μA
LVD operating current	I_{LVD} Notes 1, 7				0.08		μA
Self-programming operating current	I_{FSP} Notes 1, 9				2.50	12.20	mA
BGO operating current	I_{BGO} Notes 1, 8				2.00	12.20	mA
LCD operating current	I_{LCD1} Notes 11, 12	External resistance division method	V_{DD} = EV_{DD} = 5.0 V V_{L4} = 5.0 V		0.04	0.20	μA
	I_{LCD2} Note 11	Internal voltage boosting method	V_{DD} = EV_{DD} = 5.0 V V_{L4} = 5.1 V (VLCD = 12H)		1.12	3.70	μA
			V_{DD} = EV_{DD} = 3.0 V V_{L4} = 3.0 V (VLCD = 04H)		0.63	2.20	μA
	I_{LCD3} Note 11	Capacitor split method	V_{DD} = EV_{DD} = 3.0 V V_{L4} = 3.0 V		0.12	0.50	μA
SNOOZE operating current	I_{SNOZ} Note 1	ADC operation	The mode is performed Note 10		0.50	0.60	mA
			The A/D conversion operations are performed, Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		1.20	1.44	mA
		Simplified SPI (CSI)/UART operation			0.70	0.84	mA

(**Notes** and **Remarks** are listed on the next page.)

Notes 1. Current flowing to V_{DD}.

2. When high speed on-chip oscillator and high-speed system clock are stopped.

3. Current flowing only to the real-time clock (RTC) (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The supply current of the R7F0C001G/L, R7F0C002G/L is the sum of the values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time clock operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added. I_{DD2} subsystem clock operation includes the operational current of the real-time clock.

4. Current flowing only to the 12-bit interval timer (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The supply current of the R7F0C001G/L, R7F0C002G/L is the sum of the values of either I_{DD1} or I_{DD2}, and I_{IT}, when the 12-bit interval timer operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added.

5. Current flowing only to the watchdog timer (including the operating current of the low-speed on-chip oscillator). The supply current of the R7F0C001G/L, R7F0C002G/L is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer is in operation.

6. Current flowing only to the A/D converter. The supply current of the R7F0C001G/L, R7F0C002G/L is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

7. Current flowing only to the LVD circuit. The supply current of the R7F0C001G/L, R7F0C002G/L is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVD} when the LVD circuit is in operation.

8. Current flowing only during data flash rewrite.

9. Current flowing only during self programming.

10. For shift time to the SNOOZE mode, see **16.3.3 SNOOZE mode.**

11. Current flowing only to the LCD controller/driver. The supply current value of the R7F0C001G/L, R7F0C002G/L 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.

The TYP. value and MAX. value are following conditions.

- When f_{SUB} is selected for system clock, LCD clock = 128 Hz (LCDC0 = 07H)
- 4-Time-Slice, 1/3 Bias Method

12. Not including the current that flows through the external divider resistor when the external resistance division method is used.

Remarks 1. f_{IL}: Low-speed on-chip oscillator clock frequency

2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

3. f_{CLK}: CPU/peripheral hardware clock frequency

4. Temperature condition of the TYP. value is T_A = 25°C

27.4 AC Characteristics

27.4.1 Basic operation

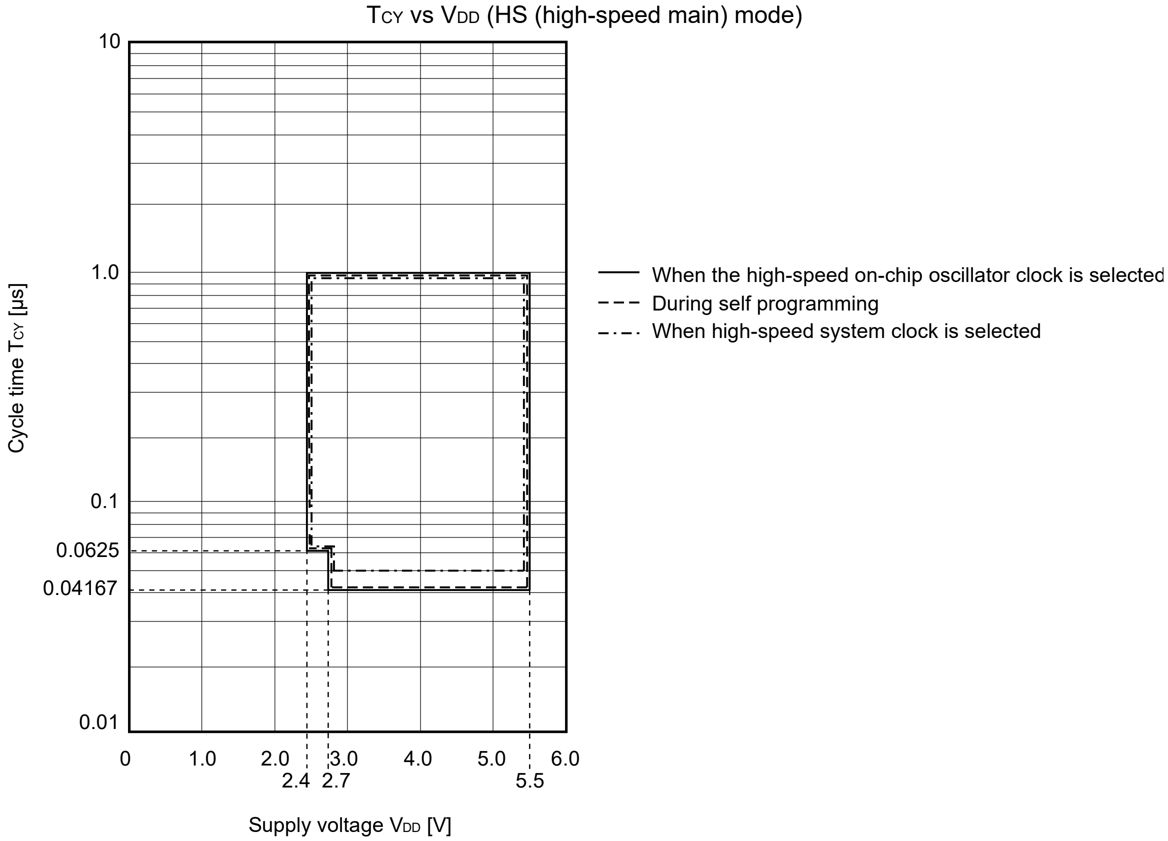
(T_A = –40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

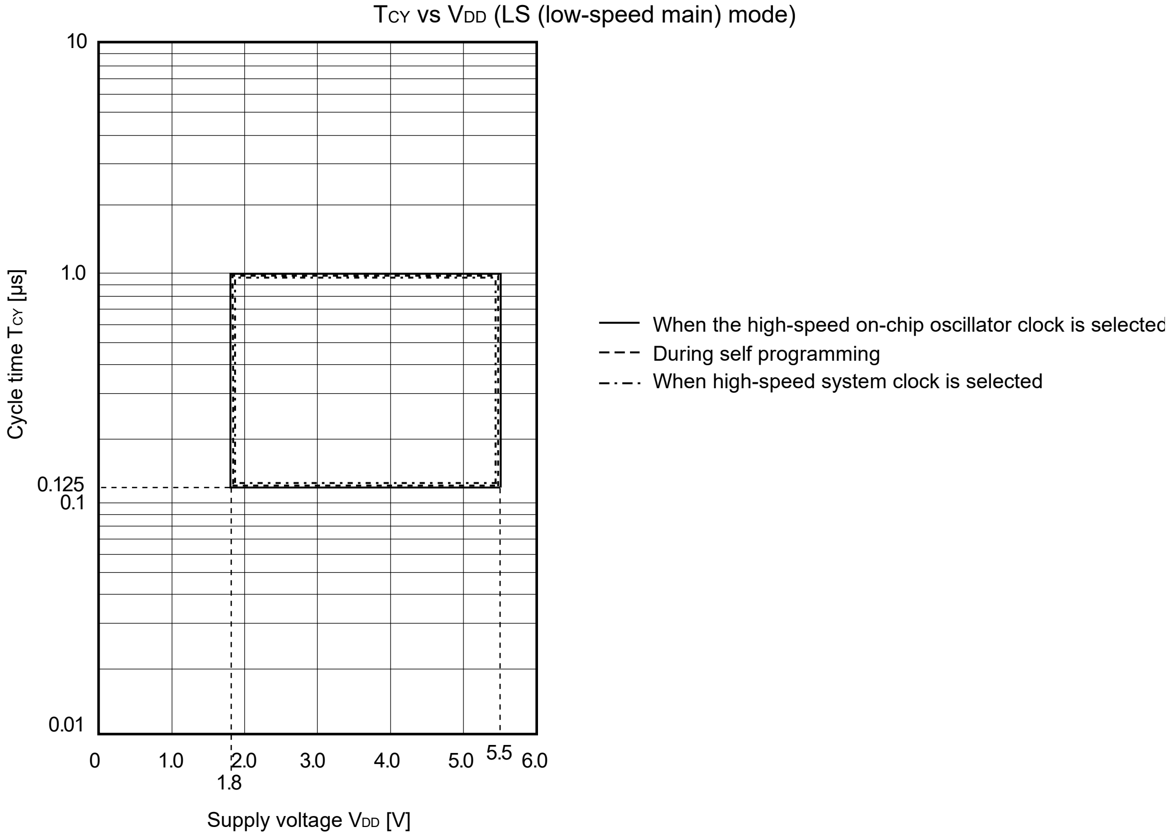
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
			LV (low voltage main) mode	1.6 V ≤ V_{DD} ≤ 5.5 V	0.25		1	μs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.125		1	μs
		Subsystem clock (f_{SUB}) operation		1.8 V ≤ V_{DD} ≤ 5.5 V	28.5	30.5	31.3	μs
		In the self programming mode	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
			LV (low voltage main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.25		1	μs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.125		1	μs
External main system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V			1.0		20.0	MHz
		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_{EXS}				32		35	kHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V			24			ns
		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 TI03, TI06, TI07, input high-level width, low-level width	t_{TIH}, t_{TIL}				$1/f_{MCK}+10$			ns
TO00 to TO03, TO06, TO07 output frequency	f_{TO}	HS (high-speed main) mode		4.0 V ≤ EV_{DD} ≤ 5.5 V			16	MHz
				2.7 V ≤ EV_{DD} < 4.0 V			8	MHz
				2.4 V ≤ EV_{DD} < 2.7 V			4	MHz
		LV (low voltage main) mode		1.6 V ≤ EV_{DD} ≤ 5.5 V			2	MHz
		LS (low-speed main) mode		1.8 V ≤ EV_{DD} ≤ 5.5 V			4	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	HS (high-speed main) mode		4.0 V ≤ EV_{DD} ≤ 5.5 V			16	MHz
				2.7 V ≤ EV_{DD} < 4.0 V			8	MHz
				2.4 V ≤ EV_{DD} < 2.7 V			4	MHz
		LV (low-voltage main) mode		1.8 V ≤ EV_{DD} ≤ 5.5 V			4	MHz
				1.6 V ≤ EV_{DD} < 1.8 V			2	MHz
		LS (low-speed main) mode		1.8 V ≤ EV_{DD} ≤ 5.5 V			4	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0		1.6 V ≤ V_{DD} ≤ 5.5 V	1			μs
		INTP1 to INTP7		1.6 V ≤ EV_{DD} ≤ 5.5 V	1			μs
Key interrupt input low-level width	t_{KR}	KR0 to KR3		1.8 V ≤ EV_{DD} ≤ 5.5 V	250			ns
				1.6 V ≤ EV_{DD} < 1.8 V	1			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

Remark f_{MCK}: Timer array unit operation clock frequency

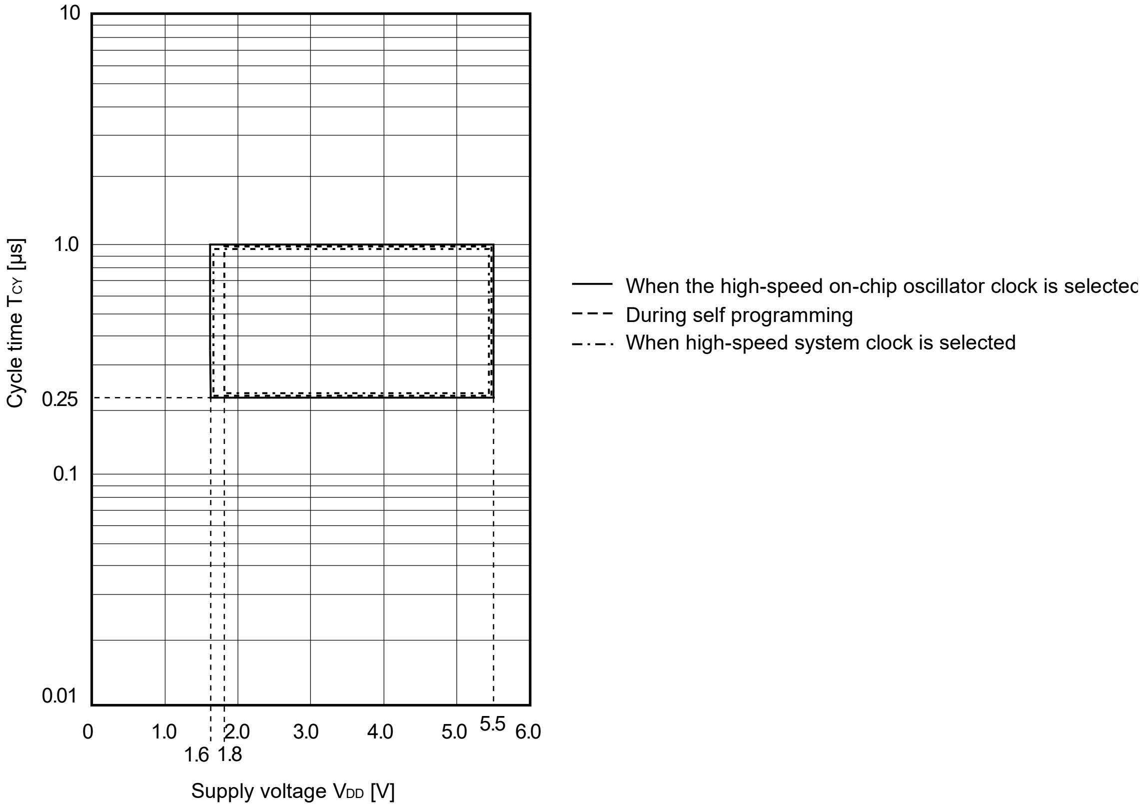
(Operation clock to be set by the CKS0n bit of timer mode register 0n (TMR0n). n: Channel number (n = 0 to 3, 6, 7))

Minimum Instruction Execution Time during Main System Clock Operation

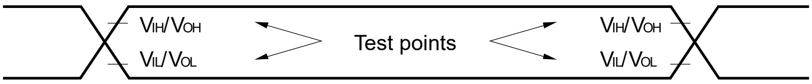




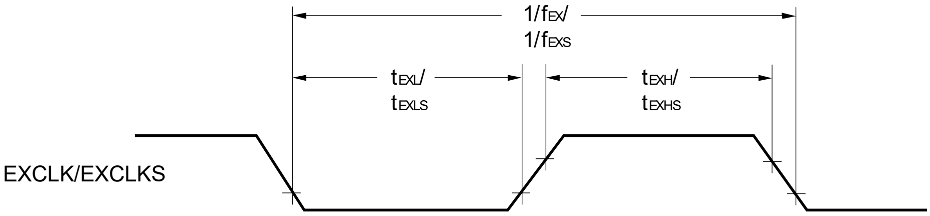
T_{CY} vs V_{DD} (LV (low-voltage main) mode)



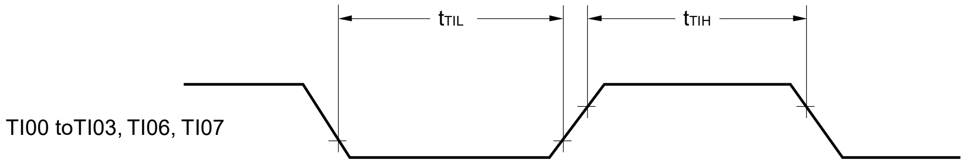
AC Timing Test Points

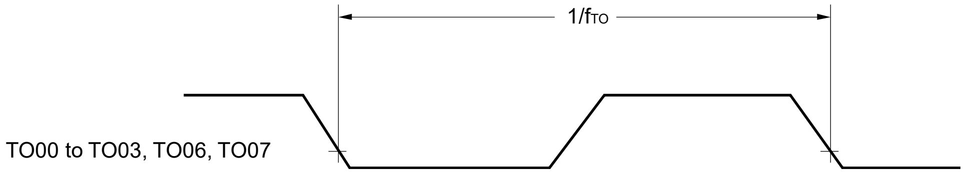


External System Clock Timing

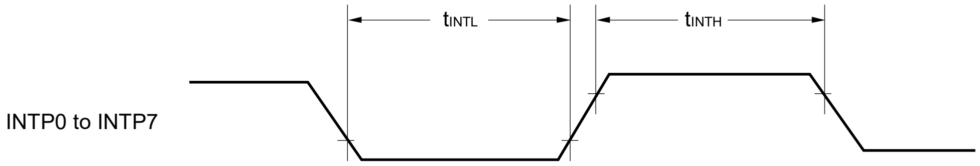


TI/TO Timing

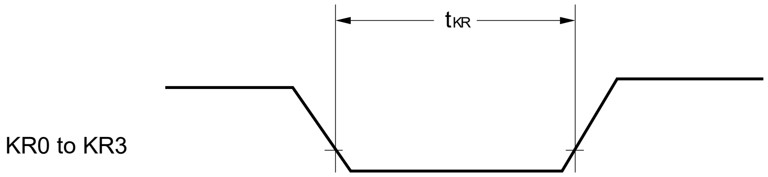




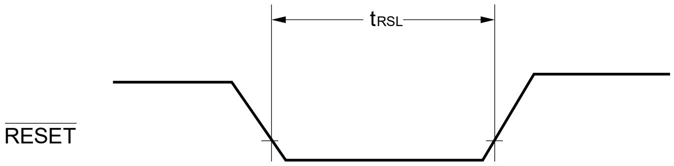
Interrupt Request Input Timing



Key Interrupt Input Timing

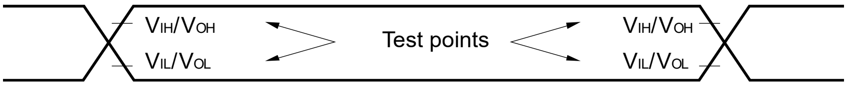


$\overline{\text{RESET}}$ Input Timing



27.5 Peripheral Functions Characteristics

AC Timing Test Points



27.5.1 Serial array unit

(1) During communication at same potential (UART mode)
(T_A = –40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{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]		2.4 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V			$f_{MCK}/6$		$f_{MCK}/6$		$f_{MCK}/6$	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} [Note 2]		4.0		1.3		0.6	Mbps
		1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V					$f_{MCK}/6$		$f_{MCK}/6$	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} [Note 2]				1.3		0.6	Mbps
		1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V							$f_{MCK}/6$	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} [Note 2]						0.6	Mbps

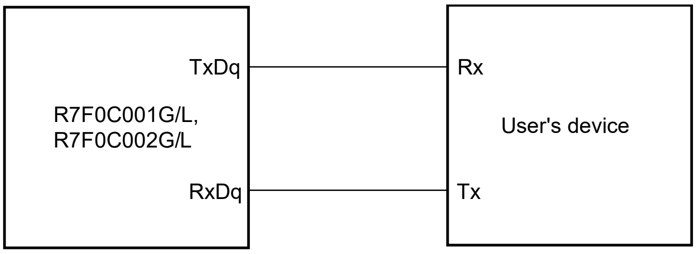
Notes 1. Transfer rate in the SNOOZE mode is 4800 bps only.

2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:

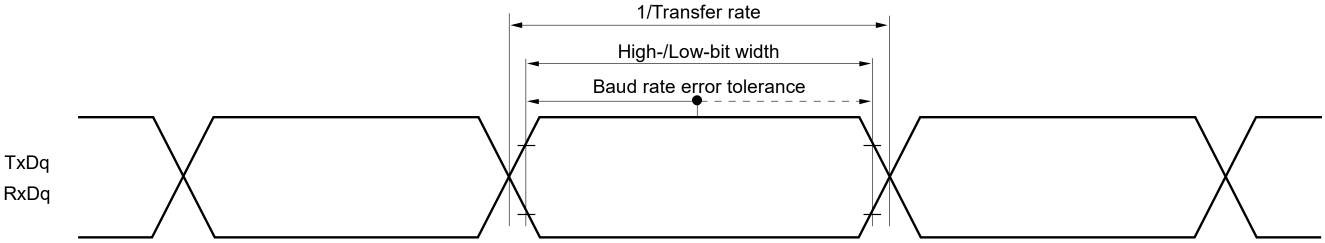
HS (high-speed main) mode:	24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
	16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode:	8 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
LV (low-voltage main) mode:	4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Remarks 1. q: UART number (q = 0), g: PIM and POM number (g = 1)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

(2) During communication at same potential (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = –40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{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}	2.7 V ≤ EV_{DD} ≤ 5.5 V	167 Note 1		500 Note 1		1000 Note 1		ns
		2.4 V ≤ EV_{DD} ≤ 5.5 V	250 Note 1		500 Note 1		1000 Note 1		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V			500 Note 1		1000 Note 1		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V					1000 Note 1		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V	t_{KCY1}/2 – 12		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
		2.7 V ≤ EV_{DD} ≤ 5.5 V	t_{KCY1}/2 – 18		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
		2.4 V ≤ EV_{DD} ≤ 5.5 V	t_{KCY1}/2 – 38		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V			t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V					t_{KCY1}/2 – 100		ns
SIp setup time (to SCKp↑) Note 2	t_{SIK1}	2.7 V ≤ EV_{DD} ≤ 5.5 V	44		110		110		ns
		2.4 V ≤ EV_{DD} ≤ 5.5 V	75		110		110		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V			110		110		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V					220		ns
SIp hold time (from SCKp↑) Note 3	t_{KSI1}	2.4 V ≤ EV_{DD} ≤ 5.5 V	19		19		19		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V			19		19		
		1.6 V ≤ EV_{DD} ≤ 5.5 V					19		
Delay time from SCKp↓ to SOp output Note 4	t_{KSO1}	C = 30 pF Note 5 2.4 V ≤ EV_{DD} ≤ 5.5 V		25		25		25	ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V				25		25	
		1.6 V ≤ EV_{DD} ≤ 5.5 V						25	

Notes **1.** For CSI00, set a cycle of 2/f_{MCK} or longer. For CSI01, set a cycle of 4/f_{MCK} or longer.

2. 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.

3. 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.

4. 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.

5. C is the load capacitance of the SCKp and SOp output lines.

Caution **Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

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

Remarks 1. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1),
g: PIM and POM numbers (g = 1)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 01))

(3) During communication at same potential (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input) (1/2)

(T_A = –40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{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}	$4.0\ V \le EV_{DD} \le 5.5\ V$	$20\ MHz < f_{MCK}$	$8/f_{MCK}$						ns
			$f_{MCK} \le 20\ MHz$	$6/f_{MCK}$		$6/f_{MCK}$		$6/f_{MCK}$		ns
		$2.7\ V \le EV_{DD} < 4.0\ V$	$16\ MHz < f_{MCK}$	$8/f_{MCK}$						ns
			$f_{MCK} \le 16\ MHz$	$6/f_{MCK}$		$6/f_{MCK}$		$6/f_{MCK}$		ns
		$2.4\ V \le EV_{DD} \le 5.5\ V$		$6/f_{MCK}$ and 500		$6/f_{MCK}$		$6/f_{MCK}$		ns
		$1.8\ V \le EV_{DD} < 2.4\ V$				$6/f_{MCK}$		$6/f_{MCK}$		ns
		$1.6\ V \le EV_{DD} < 1.8\ V$						$6/f_{MCK}$		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	$4.0\ V \le EV_{DD} \le 5.5\ V$		$t_{KCY2}/2 - 7$		$t_{KCY2}/2 - 7$		$t_{KCY2}/2 - 7$		ns
		$2.7\ V \le EV_{DD} < 4.0\ V$		$t_{KCY2}/2 - 8$		$t_{KCY2}/2 - 8$		$t_{KCY2}/2 - 8$		ns
		$2.4\ V \le EV_{DD} < 2.7\ V$		$t_{KCY2}/2 - 18$		$t_{KCY2}/2 - 18$		$t_{KCY2}/2 - 18$		ns
		$1.8\ V \le EV_{DD} < 2.4\ V$				$t_{KCY2}/2 - 18$		$t_{KCY2}/2 - 18$		ns
		$1.6\ V \le EV_{DD} < 1.8\ V$						$t_{KCY2}/2 - 66$		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}	$2.7\ V \le EV_{DD} \le 5.5\ V$		$1/f_{MCK} + 20$		$1/f_{MCK} + 30$		$1/f_{MCK} + 30$		ns
		$2.4\ V \le EV_{DD} < 2.7\ V$		$1/f_{MCK} + 30$		$1/f_{MCK} + 30$		$1/f_{MCK} + 30$		
		$1.8\ V \le EV_{DD} < 2.4\ V$				$1/f_{MCK} + 30$		$1/f_{MCK} + 30$		ns
		$1.6\ V \le EV_{DD} < 1.8\ V$						$1/f_{MCK} + 40$		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}	$2.4\ V \le EV_{DD} \le 5.5\ V$		$1/f_{MCK} + 31$		$1/f_{MCK} + 31$		$1/f_{MCK} + 31$		ns
		$1.8\ V \le EV_{DD} < 2.4\ V$				$1/f_{MCK} + 31$		$1/f_{MCK} + 31$		ns
		$1.6\ V \le EV_{DD} < 1.8\ V$						$1/f_{MCK} + 250$		ns

(**Notes**, **Caution,** and **Remarks** are listed on the next page.)

(3) During communication at same potential Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input) (2/2)

(T_A = –40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode	LS (low-speed main) Mode	LV (low-voltage main) Mode	Unit	Para meter	Symbol	Conditions
Delay time from SCKp↓ to SOp output [Note 3]	t_{KSO2}	C = 30 pF [Note 4]	4.0 V ≤ EV_{DD} ≤ 5.5 V		$2/f_{MCK}$ + 44		$2/f_{MCK}$ + 110		$2/f_{MCK}$ + 110	ns
			2.7 V ≤ EV_{DD} < 4.0 V		$2/f_{MCK}$ + 44		$2/f_{MCK}$ + 110		$2/f_{MCK}$ + 110	ns
			2.4 V ≤ EV_{DD} < 2.7 V		$2/f_{MCK}$ + 75		$2/f_{MCK}$ + 110		$2/f_{MCK}$ + 110	ns
			1.8 V ≤ EV_{DD} < 2.4 V				$2/f_{MCK}$ + 110		$2/f_{MCK}$ + 110	ns
			1.6 V ≤ EV_{DD} < 1.8 V						$2/f_{MCK}$ + 220	ns

Notes

1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

5. Transfer rate in the SNOOZE mode : MAX. 1 Mbps

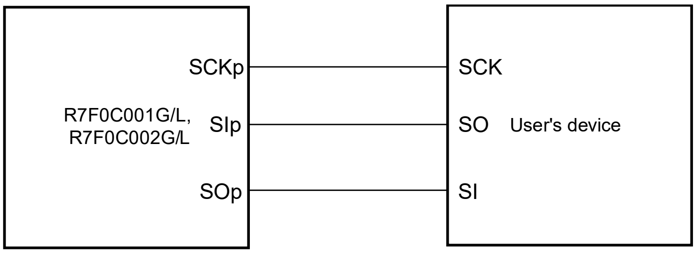
Caution Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks

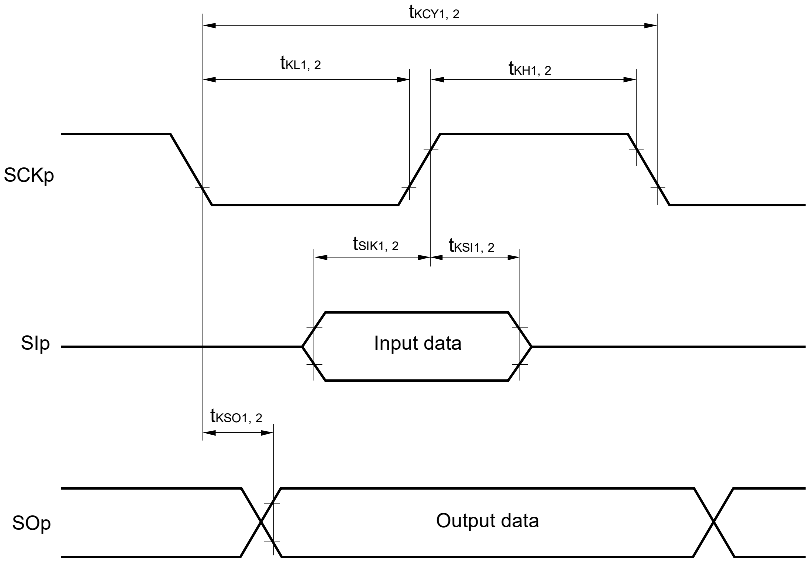
1. p: CSI number (p = 00, 01), m: Unit number (m = 0),
n: Channel number (n = 0, 1), g: PIM number (g = 1)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

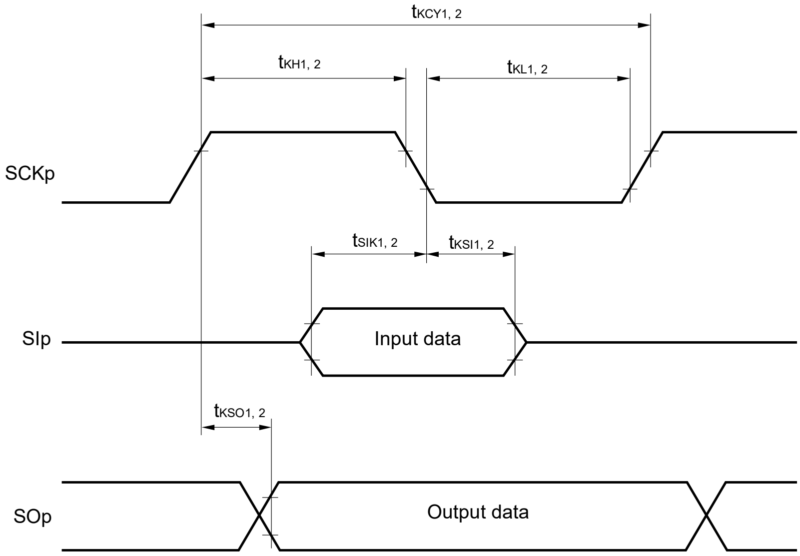
Simplified SPI (CSI) mode connection diagram (during communication at same potential)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remarks 1. p: CSI number (p = 00, 01)

2. m: Unit number, n: Channel number (mn = 00, 01)

(4) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode) (1/2)

(T_A = –40 to +85°C, 1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{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		Reception	$4.0\ V \le EV_{DD} \le 5.5\ V$, $2.7\ V \le V_b \le 4.0\ V$			$f_{MCK}/6$ Note 1		$f_{MCK}/6$ Note 1		$f_{MCK}/6$ Note 1	bps
				Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Note 3		4.0		1.3		0.6	Mbps
			$2.7\ V \le EV_{DD} < 4.0\ V$, $2.3\ V \le V_b \le 2.7\ V$			$f_{MCK}/6$ 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 3		4.0		1.3		0.6	Mbps
			$2.4\ V \le EV_{DD} < 3.3\ V$, $1.6\ V \le V_b \le 2.0\ V$			$f_{MCK}/6$ Note 1		$f_{MCK}/6$ Note 1		$f_{MCK}/6$ Note 1	bps
				Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Note 3		4.0		1.3		0.6	Mbps
			$1.8\ V \le EV_{DD} < 3.3\ V$, $1.6\ V \le V_b \le 2.0\ V$					$f_{MCK}/6$ Notes 1, 2		$f_{MCK}/6$ Notes 1, 2	bps
				Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Note 3				1.3		0.6	Mbps

Notes 1. Transfer rate in the SNOOZE mode is 4800 bps only.

2. Use it with $EV_{DD} \ge V_b$.

3. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:

HS (high-speed main) mode:	24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
	16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode:	8 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
LV (low-voltage main) mode:	4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance (48-pin products)/EV_{DD} tolerance (64-pin products)) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remarks 1. V_b[V]: Communication line voltage

2. q: UART number (q = 0), g: PIM and POM number (g = 1)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01)

(4) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode) (2/2)

(T_A = –40 to +85°C, 1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{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		Transmission	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V			**Note 1**		**Note 1**		**Note 1**	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V		2.8 Note 2		2.8 Note 2		2.8 Note 2	Mbps
			2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V			**Note 3**		**Note 3**		**Note 3**	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ V_b = 2.3 V		1.2 Note 4		1.2 Note 4		1.2 Note 4	Mbps
			2.4 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V			**Note 6**		**Note 6**		**Note 6**	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ V_b = 1.6 V		0.43 Note 7		0.43 Note 7		0.43 Note 7	Mbps
			1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V					**Notes 5, 6**		**Notes 5, 6**	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V				0.43 Note 7		0.43 Note 7	Mbps

Notes 1. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V ≤ EV_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln\left(1 - \frac{2.2}{V_b}\right)\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln\left(1 - \frac{2.2}{V_b}\right)\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

2. This value as an example is calculated when the conditions described in the "Conditions" column are met. Refer to Note 1 above to calculate the maximum transfer rate under conditions of the customer.

3. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.

   Expression for calculating the transfer rate when $2.7\ V \leq EV_{DD} < 4.0\ V$ and $2.3\ V \leq V_b \leq 2.7\ V$

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln (1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln (1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

   * This value is the theoretical value of the relative difference between the transmission and reception sides.

4. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to Note 3 above to calculate the maximum transfer rate under conditions of the customer.
5. Use it with $EV_{DD} \geq V_b$.
6. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.

   Expression for calculating the transfer rate when $1.8\ V \leq EV_{DD} < 3.3\ V$ and $1.6\ V \leq V_b \leq 2.0\ V$

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln (1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

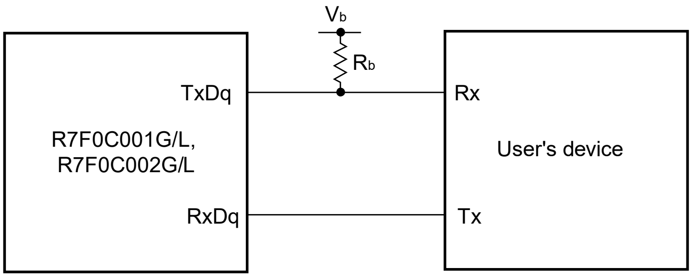
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln (1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

   * This value is the theoretical value of the relative difference between the transmission and reception sides.

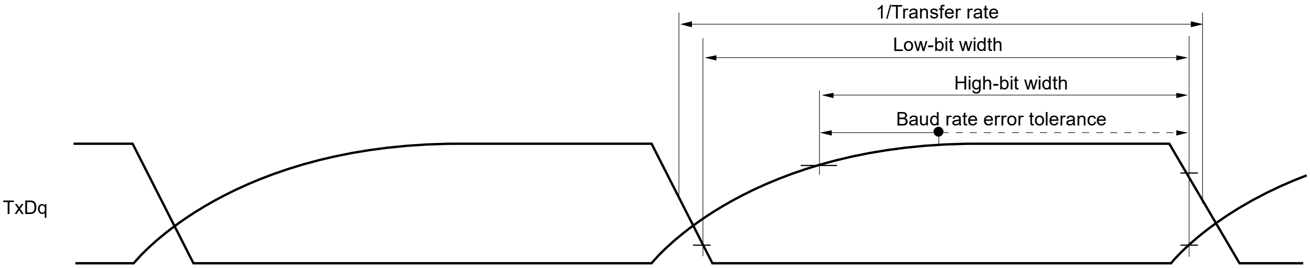
7. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to Note 6 above to calculate the maximum transfer rate under conditions of the customer.

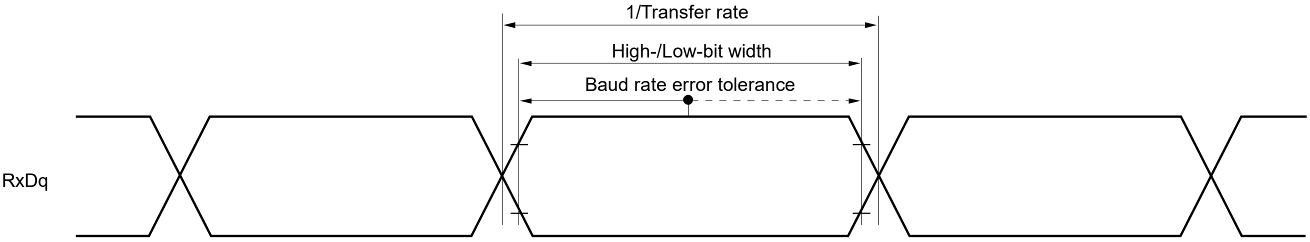
Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance (48-pin products)/EV_{DD} tolerance (64-pin products)) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)





Remarks 1. R_b[Ω]:Communication line (TxDq) pull-up resistance,
C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

2. q: UART number (q = 0, 1), g: PIM and POM number (g = 1)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

(5) Communication at different potential (2.5 V, 3 V) Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output, corresponding CSI00 only)

(T_A = –40 to +85°C, 2.7 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{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}	$t_{KCY1} \ge 2/f_{CLK}$	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	200 Note 1		1150 Note 1		1150 Note 1		ns
			2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	300 Note 1		1150 Note 1		1150 Note 1		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		$t_{KCY1}/2 - 50$		$t_{KCY1}/2 - 50$		$t_{KCY1}/2 - 50$		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		$t_{KCY1}/2 - 120$		$t_{KCY1}/2 - 120$		$t_{KCY1}/2 - 120$		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		$t_{KCY1}/2 - 7$		$t_{KCY1}/2 - 50$		$t_{KCY1}/2 - 50$		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		$t_{KCY1}/2 - 10$		$t_{KCY1}/2 - 50$		$t_{KCY1}/2 - 50$		ns
SIp setup time (to SCKp↑) Note 2	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		58		479		479		ns
		2.7 V ≤ EV_{DD} < 4.0 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 2	t_{KSI1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		10		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		10		ns
Delay time from SCKp↓ to SOp output Note 2	t_{KSO1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ			60		60		60	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			130		130		130	ns
SIp setup time (to SCKp↓) Note 3	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		23		110		110		ns
		2.7 V ≤ EV_{DD} < 4.0 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 3	t_{KSI1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		10		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		10		ns
Delay time from SCKp↑ to SOp output Note 3	t_{KSO1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ			10		10		10	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			10		10		10	ns

(**Notes**, **Caution** and **Remarks** are listed on the next page.)

Notes 1. Set a cycle of 2/f_{MCK} or longer.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

3. 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 (48-pin products)/EV_{DD} tolerance (64-pin products)) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remarks 1. R_b[Ω]:Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

2. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1),
g: PIM and POM number (g = 1)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01)

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output) (1/3)

(T_A = –40 to +85°C, 1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{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} ≥ 4/f_{CLK}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	300		1150		1150		ns
			2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	500		1150		1150		ns
			2.4 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	1150		1150		1150		ns
			1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V [Note], C_b = 30 pF, R_b = 5.5 kΩ			1150		1150		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 75		t_{KCY1}/2 – 75		t_{KCY1}/2 – 75		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 170		t_{KCY1}/2 – 170		t_{KCY1}/2 – 170		ns
		2.4 V ≤ EV_{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
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V [Note], C_b = 30 pF, R_b = 5.5 kΩ				t_{KCY1}/2 – 458		t_{KCY1}/2 – 458		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 12		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 18		t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns
		2.4 V ≤ EV_{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
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V [Note], C_b = 30 pF, R_b = 5.5 kΩ				t_{KCY1}/2 – 50		t_{KCY1}/2 – 50		ns

Note Use it with EV_{DD} ≥ V_b.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance (48-pin products)/EV_{DD} tolerance (64-pin products)) 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.

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output) (2/3)

(T_A = –40 to +85°C, 1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{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.	
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	81		479		479		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	177		479		479		ns
		2.4 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	479		479		479		ns
		1.8 V ≤ EV_{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		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		19		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		ns
		2.4 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	19		19		19		ns
		1.8 V ≤ EV_{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		ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		100		100		100	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		195		195		195	ns
		2.4 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		483		483		483	ns
		1.8 V ≤ EV_{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	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	44		110		110		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	44		110		110		ns
		2.4 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	110		110		110		ns
		1.8 V ≤ EV_{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		ns

Notes

1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.
2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.
3. Use it with EV_{DD} ≥ V_b.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance (48-pin products)/EV_{DD} tolerance (64-pin products)) 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.

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output) (3/3)

(T_A = –40 to +85°C, 1.8 V $\leq$ EV_{DD} = V_{DD} $\leq$ 5.5 V, V_{SS} = EV_{SS} = 0 V)

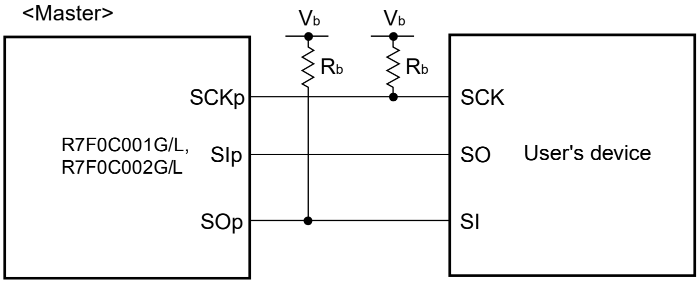
Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	4.0 V $\leq$ EV_{DD} $\leq$ 5.5 V, 2.7 V $\leq$ V_b $\leq$ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		19		ns
		2.7 V $\leq$ EV_{DD} < 4.0 V, 2.3 V $\leq$ V_b $\leq$ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		ns
		2.4 V $\leq$ EV_{DD} < 3.3 V, 1.6 V $\leq$ V_b $\leq$ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	19		19		19		ns
		1.8 V $\leq$ EV_{DD} < 3.3 V, 1.6 V $\leq$ V_b $\leq$ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ			19		19		ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	4.0 V $\leq$ EV_{DD} $\leq$ 5.5 V, 2.7 V $\leq$ V_b $\leq$ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		25		25		25	ns
		2.7 V $\leq$ EV_{DD} < 4.0 V, 2.3 V $\leq$ V_b $\leq$ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		25		25		25	ns
		2.4 V $\leq$ EV_{DD} < 3.3 V, 1.6 V $\leq$ V_b $\leq$ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		25		25		25	ns
		1.8 V $\leq$ EV_{DD} < 3.3 V, 1.6 V $\leq$ V_b $\leq$ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ				25		25	ns

Notes

1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.
2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.
3. Use it with $EV_{DD} \geq V_b$.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance (48-pin products)/EV_{DD} tolerance (64-pin products)) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Simplified SPI (CSI) mode connection diagram (during communication at different potential)



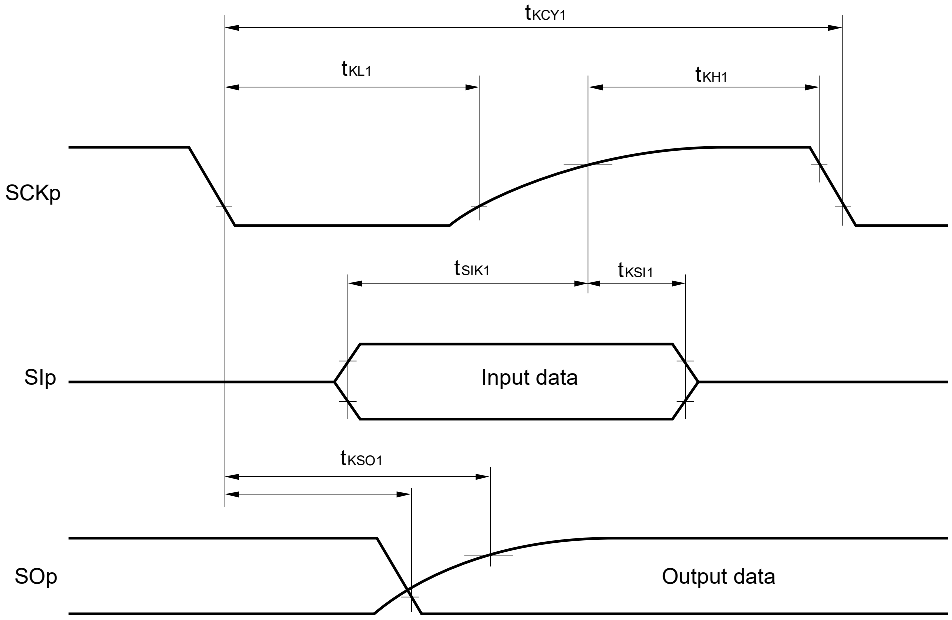
Remarks 1. R_b[Ω]:Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

2. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM number (g = 1)

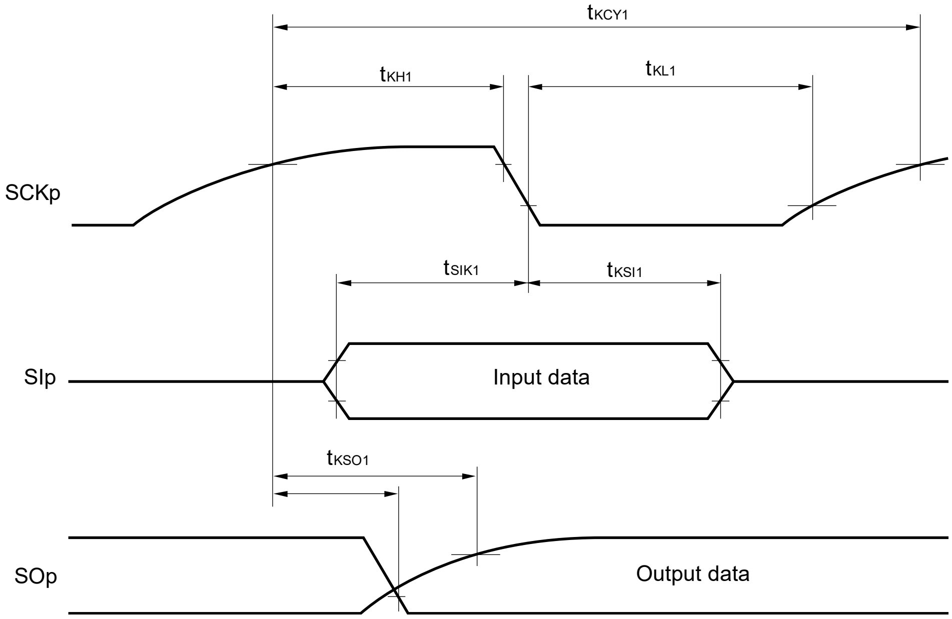
3. f_{MCK}: Serial array unit operation clock frequency

(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01)

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, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1),
g: PIM and POM number (g = 1)

(7) Communication at different potential (1.8 V, 2.5 V, 3 V) Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = –40 to +85°C, 1.8 V ≤ $EV_{DD} = V_{DD}$ ≤ 5.5 V, $V_{SS} = EV_{SS}$ = 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 Note 1	t_{KCY2}	$4.0\ V \le EV_{DD} \le 5.5\ V$, $2.7\ V \le V_b \le 4.0\ V$	$20\ MHz < f_{MCK} \le 24\ MHz$	$12/f_{MCK}$						ns
			$8\ MHz < f_{MCK} \le 20\ MHz$	$10/f_{MCK}$						ns
			$4\ MHz < f_{MCK} \le 8\ MHz$	$8/f_{MCK}$		$16/f_{MCK}$				ns
			$f_{MCK} \le 4\ MHz$	$6/f_{MCK}$		$10/f_{MCK}$		$10/f_{MCK}$		ns
		$2.7\ V \le EV_{DD} < 4.0\ V$, $2.3\ V \le V_b \le 2.7\ V$	$20\ MHz < f_{MCK} \le 24\ MHz$	$16/f_{MCK}$						ns
			$16\ MHz < f_{MCK} \le 20\ MHz$	$14/f_{MCK}$						ns
			$8\ MHz < f_{MCK} \le 16\ MHz$	$12/f_{MCK}$						ns
			$4\ MHz < f_{MCK} \le 8\ MHz$	$8/f_{MCK}$		$16/f_{MCK}$				ns
			$f_{MCK} \le 4\ MHz$	$6/f_{MCK}$		$10/f_{MCK}$		$10/f_{MCK}$		ns
		$2.4\ V \le EV_{DD} < 3.3\ V$, $1.6\ V \le V_b \le 2.0\ V$	$20\ MHz < f_{MCK} \le 24\ MHz$	$36/f_{MCK}$						ns
			$16\ MHz < f_{MCK} \le 20\ MHz$	$32/f_{MCK}$						ns
			$8\ MHz < f_{MCK} \le 16\ MHz$	$26/f_{MCK}$						ns
			$4\ MHz < f_{MCK} \le 8\ MHz$	$16/f_{MCK}$		$16/f_{MCK}$				ns
			$f_{MCK} \le 4\ MHz$	$10/f_{MCK}$		$10/f_{MCK}$		$10/f_{MCK}$		ns
		$1.8\ V \le EV_{DD} < 3.3\ V$, $1.6\ V \le V_b \le 2.0\ V$ Note 2	$4\ MHz < f_{MCK} \le 8\ MHz$			$16/f_{MCK}$				ns
			$f_{MCK} \le 4\ MHz$			$10/f_{MCK}$		$10/f_{MCK}$		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	$4.0\ V \le EV_{DD} \le 5.5\ V$, $2.7\ V \le V_b \le 4.0\ V$		$t_{KCY2}/2 - 12$		$t_{KCY2}/2 - 50$		$t_{KCY2}/2 - 50$		ns
		$2.7\ V \le EV_{DD} < 4.0\ V$, $2.3\ V \le V_b \le 2.7\ V$		$t_{KCY2}/2 - 18$		$t_{KCY2}/2 - 50$		$t_{KCY2}/2 - 50$		ns
		$2.4\ V \le EV_{DD} < 3.3\ V$, $1.6\ V \le V_b \le 2.0\ V$		$t_{KCY2}/2 - 50$		$t_{KCY2}/2 - 50$		$t_{KCY2}/2 - 50$		ns
		$1.8\ V \le EV_{DD} < 3.3\ V$, $1.6\ V \le V_b \le 2.0\ V$ Note 2				$t_{KCY2}/2 - 50$		$t_{KCY2}/2 - 50$		ns
SIp setup time (to SCKp↑) Note 3	t_{SIK2}	$4.0\ V \le EV_{DD} < 5.5\ V$, $2.7\ V \le V_b \le 4.0\ V$		$1/f_{MCK} + 20$		$1/f_{MCK} + 30$		$1/f_{MCK} + 30$		ns
		$2.7\ V \le EV_{DD} < 4.0\ V$, $2.3\ V \le V_b \le 2.7\ V$		$1/f_{MCK} + 20$		$1/f_{MCK} + 30$		$1/f_{MCK} + 30$		ns
		$2.4\ V \le EV_{DD} < 3.3\ V$, $1.6\ V \le V_b \le 2.0\ V$		$1/f_{MCK} + 30$		$1/f_{MCK} + 30$		$1/f_{MCK} + 30$		ns
		$1.8\ V \le EV_{DD} < 3.3\ V$, $1.6\ V \le V_b \le 2.0\ V$ Note 2				$1/f_{MCK} + 30$		$1/f_{MCK} + 30$		ns
SIp hold time (from SCKp↑) Note 4	t_{KSI2}	$4.0\ V \le EV_{DD} < 5.5\ V$, $2.7\ V \le V_b \le 4.0\ V$		$1/f_{MCK} + 31$		$1/f_{MCK} + 31$		$1/f_{MCK} + 31$		ns
		$2.7\ V \le EV_{DD} < 4.0\ V$, $2.3\ V \le V_b \le 2.7\ V$		$1/f_{MCK} + 31$		$1/f_{MCK} + 31$		$1/f_{MCK} + 31$		ns
		$2.4\ V \le EV_{DD} < 3.3\ V$, $1.6\ V \le V_b \le 2.0\ V$		$1/f_{MCK} + 31$		$1/f_{MCK} + 31$		$1/f_{MCK} + 31$		ns
		$1.8\ V \le EV_{DD} < 3.3\ V$, $1.6\ V \le V_b \le 2.0\ V$ Note 2				$1/f_{MCK} + 31$		$1/f_{MCK} + 31$		ns

(**Notes**, **Caution** and **Remarks** are listed on the next page.)

(7) Communication at different potential (1.8 V, 2.5 V, 3 V) Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = –40 to +85°C, 1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{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.	
Delay time from SCKp↓ to SOp output Note 5	t_{KSO2}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		2/f_{MCK} + 120		2/f_{MCK} + 573		2/f_{MCK} + 573	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		2/f_{MCK} + 214		2/f_{MCK} + 573		2/f_{MCK} + 573	ns
		2.4 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		2/f_{MCK} + 573		2/f_{MCK} + 573		2/f_{MCK} + 573	ns
		1.8 V ≤ EV_{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	ns

Notes 1. Transfer rate in the SNOOZE mode : MAX. 1 Mbps

2. Use it with EV_{DD} ≥ V_b.

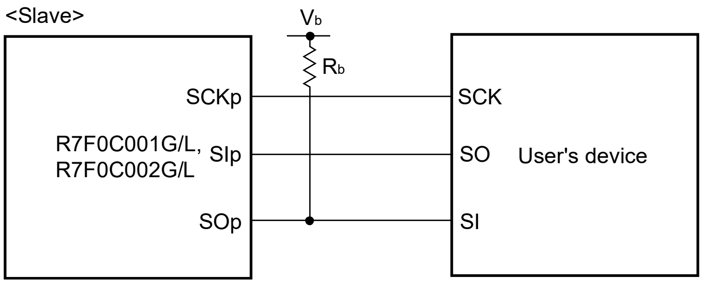
3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

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.

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 (48-pin products)/EV_{DD} tolerance (64-pin products)) 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.

Simplified SPI (CSI) mode connection diagram (during communication at different potential)

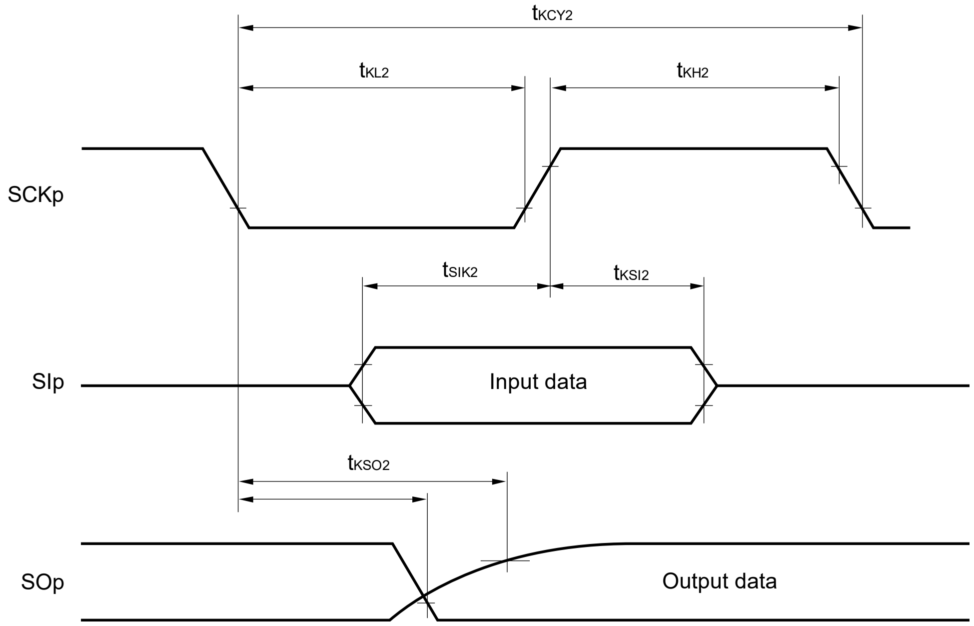


Remarks 1. R_b[Ω]:Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

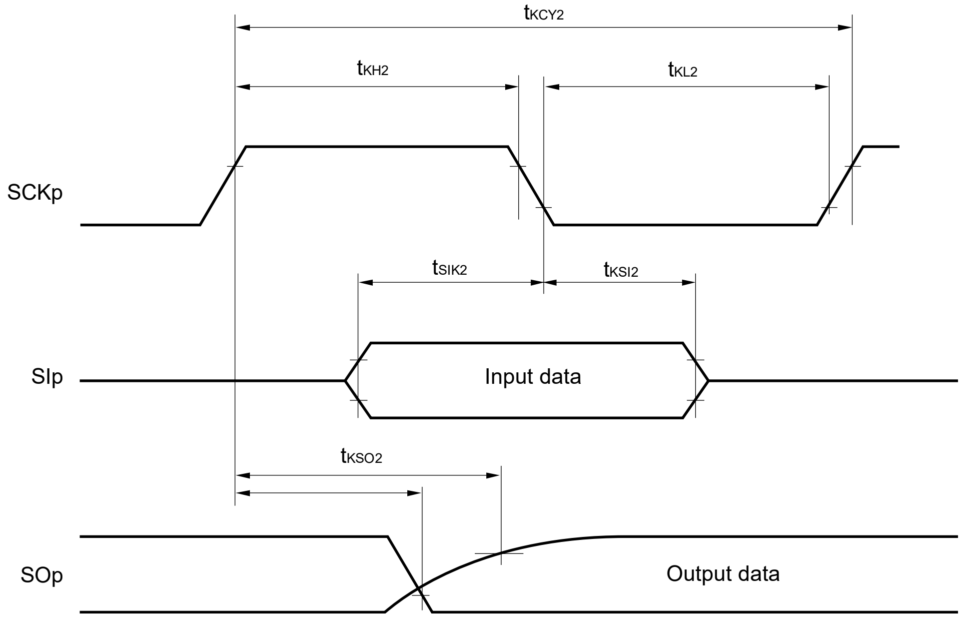
2. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM number (g = 1)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

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, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM number (g = 1)

27.6 Analog Characteristics

27.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Input channel	Reference Voltage		
	Reference voltage (+) = AV_{REFP} Reference voltage (–) = AV_{REFM}	Reference voltage (+) = V_{DD} Reference voltage (–) = V_{SS}	Reference voltage (+) = V_{BGR} Reference voltage (–) = AV_{REFM}
ANI0, ANI1	–	Refer to **27.6.1 (3)**.	Refer to **27.6.1 (4)**.
ANI16 to ANI23	Refer to **27.6.1 (2)**.		
Internal reference voltage Temperature sensor output voltage	Refer to **27.6.1 (1)**.		–

(1) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin : internal reference voltage, and temperature sensor output voltage

(T_A = –40 to +85°C, 2.4 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, 1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1	AINL	10-bit resolution AV_{REFP} = V_{DD} Note 3	1.8 V ≤ V_{DD} ≤ 5.5 V		1.2	±3.5	LSB
			1.6 V ≤ V_{DD} ≤ 5.5 V Note 4		1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error Notes 1, 2	E_{ZS}	10-bit resolution AV_{REFP} = V_{DD} Note 3	1.8 V ≤ AV_{REFP} ≤ 5.5 V			±0.25	%FSR
			1.6 V ≤ AV_{REFP} ≤ 5.5 V Note 4			±0.50	%FSR
Full-scale error Notes 1, 2	E_{FS}	10-bit resolution AV_{REFP} = V_{DD} Note 3	1.8 V ≤ AV_{REFP} ≤ 5.5 V			±0.25	%FSR
			1.6 V ≤ AV_{REFP} ≤ 5.5 V Note 4			±0.50	%FSR
Integral linearity error Note 1	ILE	10-bit resolution AV_{REFP} = V_{DD} Note 3	1.8 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
			1.6 V ≤ V_{DD} ≤ 5.5 V Note 4			±5.0	LSB
Differential linearity error Note 1	DLE	10-bit resolution AV_{REFP} = V_{DD} Note 3	1.8 V ≤ V_{DD} ≤ 5.5 V			±1.5	LSB
			1.6 V ≤ V_{DD} ≤ 5.5 V Note 4			±2.0	LSB
Analog input voltage	V_{AIN}	Internal reference voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{BGR} Note 5			V
	V_{BGR}	Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{TMPS25} Note 5			V

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

Notes **1.** Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When $AV_{REFP} < V_{DD}$, the MAX. values are as follows.
Overall error: Add ±1.0 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.
Zero-scale error/Full-scale error: Add ±0.05%FSR to the MAX. value when $AV_{REFP} = V_{DD}$.
Integral linearity error/Differential linearity error: Add ±0.5 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.

4. Values when the conversion time is set to 57 μs (min.) and 95 μs (max.).

5. Refer to **27.6.2 Temperature sensor/internal reference voltage characteristics**.

(2) When reference voltage (+) = AV$_{REFP}$/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV$_{REFM}$/ANI1 (ADREFM = 1), target pin : ANI16 to ANI23

(T$_A$ = –40 to +85°C, 1.6 V ≤ EV$_{DD}$ = V$_{DD}$ ≤ 5.5 V, 1.6 V ≤ AV$_{REFP}$ ≤ V$_{DD}$ ≤ 5.5 V, V$_{SS}$ = EV$_{SS}$ = 0 V, Reference voltage (+) = AV$_{REFP}$, Reference voltage (–) = AV$_{REFM}$ = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution AV$_{REFP}$ = EV$_{DD}$ = V$_{DD}$ Note 3	1.8 V ≤ AV$_{REFP}$ ≤ 5.5 V		1.2	±5.0	LSB
			1.6 V ≤ AV$_{REFP}$ ≤ 5.5 V Note 4		1.2	±8.5	LSB
Conversion time	t$_{CONV}$	10-bit resolution AV$_{REFP}$ = EV$_{DD}$ = V$_{DD}$ Note 3	3.6 V ≤ V$_{DD}$ ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V$_{DD}$ ≤ 5.5 V	3.1875		39	μs
			1.8 V ≤ V$_{DD}$ ≤ 5.5 V	17		39	μs
			1.6 V ≤ V$_{DD}$ ≤ 5.5 V	57		95	μs
Zero-scale error[Notes 1, 2]	E$_{ZS}$	10-bit resolution AV$_{REFP}$ = EV$_{DD}$ = V$_{DD}$ Note 3	1.8 V ≤ AV$_{REFP}$ ≤ 5.5 V			±0.35	%FSR
			1.6 V ≤ AV$_{REFP}$ ≤ 5.5 V Note 4			±0.60	%FSR
Full-scale error[Notes 1, 2]	E$_{FS}$	10-bit resolution AV$_{REFP}$ = EV$_{DD}$ = V$_{DD}$ Note 3	1.8 V ≤ AV$_{REFP}$ ≤ 5.5 V			±0.35	%FSR
			1.6 V ≤ AV$_{REFP}$ ≤ 5.5 V Note 4			±0.60	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution AV$_{REFP}$ = EV$_{DD}$ = V$_{DD}$ Note 3	1.8 V ≤ AV$_{REFP}$ ≤ 5.5 V			±3.5	LSB
			1.6 V ≤ AV$_{REFP}$ ≤ 5.5 V Note 4			±6.0	LSB
Differential linearity error Note 1	DLE	10-bit resolution AV$_{REFP}$ = EV$_{DD}$ = V$_{DD}$ Note 3	1.8 V ≤ AV$_{REFP}$ ≤ 5.5 V			±2.0	LSB
			1.6 V ≤ AV$_{REFP}$ ≤ 5.5 V Note 4			±2.5	LSB
Analog input voltage	V$_{AIN}$			0		AV$_{REFP}$ and EV$_{DD}$	V

Notes **1.** Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When AV$_{REFP}$ < EV$_{DD}$ = V$_{DD}$, the MAX. values are as follows.
Overall error: Add ±4.0 LSB to the MAX. value when AV$_{REFP}$ = V$_{DD}$.
Zero-scale error/Full-scale error: Add ±0.20%FSR to the MAX. value when AV$_{REFP}$ = V$_{DD}$.
Integral linearity error/Differential linearity error: Add ±2.0 LSB to the MAX. value when AV$_{REFP}$ = V$_{DD}$.

4. When the conversion time is set to 57 μs (min.) and 95 μs (max.).

(3) When reference voltage (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (–) = V_{SS} (ADREFM = 0), target pin : ANI0, ANI1, ANI16 to ANI23, internal reference voltage, and temperature sensor output voltage

(T_A = –40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V, Reference voltage (+) = V_{DD}, Reference voltage (–) = V_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V		1.2	±7.0	LSB
			1.6 V ≤ V_{DD} ≤ 5.5 V [Note 3]		1.2	±10.5	LSB
Conversion time	t_{CONV}	10-bit resolution	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			1.8 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
			1.6 V ≤ V_{DD} ≤ 5.5 V	57		95	μs
		10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error[Notes 1, 2]	E_{ZS}	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
			1.6 V ≤ V_{DD} ≤ 5.5 V [Note 3]			±0.85	%FSR
Full-scale error[Notes 1, 2]	E_{FS}	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
			1.6 V ≤ V_{DD} ≤ 5.5 V [Note 3]			±0.85	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±4.0	LSB
			1.6 V ≤ V_{DD} ≤ 5.5 V [Note 3]			±6.5	LSB
Differential linearity error[Note 1]	DLE	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
			1.6 V ≤ V_{DD} ≤ 5.5 V [Note 3]			±2.5	LSB
Analog input voltage	V_{AIN}	ANI0, ANI1		0		V_{DD}	V
		ANI16 to ANI23		0		EV_{DD}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{BGR}[Note 4]			V
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{TMPS25}[Note 4]			V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When the conversion time is set to 57 μs (min.) and 95 μs (max.).

4. Refer to **27.6.2 Temperature sensor/internal reference voltage characteristics**.

(4) When reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin : ANI0, ANI16 to ANI23

(T_A = –40 to +85°C, 2.4 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V, Reference voltage (+) = V_{BGR} Note 3, Reference voltage (–) = AV_{REFM} Note 4 = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8			bit
Conversion time	t_{CONV}	8-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error Notes 1, 2	E_{ZS}	8-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
Integral linearity error Note 1	ILE	8-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
Differential linearity error Note 1	DLE	8-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±1.0	LSB
Analog input voltage	V_{AIN}			0		V_{BGR} Note 3	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. Refer to **27.6.2 Temperature sensor/internal reference voltage characteristics**.

4. When reference voltage (–) = V_{SS}, the MAX. values are as follows.
Zero-scale error: Add ±0.35%FSR to the MAX. value when reference voltage (–) = AV_{REFM}.
Integral linearity error: Add ±0.5 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.
Differential linearity error: Add ±0.2 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.

27.6.2 Temperature sensor/internal reference voltage characteristics

(T_A = –40 to +85°C, 2.4 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V) (HS (high-speed main) mode)

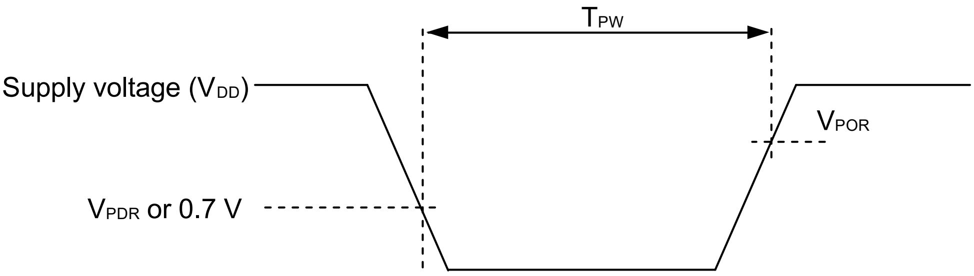
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Internal reference voltage	V_{BGR}	Setting ADS register = 81H	1.38	1.45	1.5	V
Temperature coefficient	F_{VTMPS}	Temperature sensor that depends on the temperature		–3.6		mV/°C
Operation stabilization wait time	t_{AMP}		5			μs

27.6.3 POR circuit characteristics

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.47	1.51	1.55	V
	V_{PDR}	Power supply fall time	1.46	1.50	1.54	V
Minimum pulse width[Note]	T_{PW}		300			μs

Note Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}. This is also the minimum time required for a POR reset from when V_{DD} exceeds below 0.7 V to when V_{DD} exceeds V_{POR} while STOP mode is entered or the main system clock is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



27.6.4 LVD circuit characteristics

(T_A = –40 to +85°C, $V_{PDR} \leq EV_{DD} = V_{DD} \leq 5.5$ V, $V_{SS} = EV_{SS} = 0$ V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD0}	Power supply rise time	3.98	4.06	4.14	V
			Power supply fall time	3.90	3.98	4.06	V
		V_{LVD1}	Power supply rise time	3.68	3.75	3.82	V
			Power supply fall time	3.60	3.67	3.74	V
		V_{LVD2}	Power supply rise time	3.07	3.13	3.19	V
			Power supply fall time	3.00	3.06	3.12	V
		V_{LVD3}	Power supply rise time	2.96	3.02	3.08	V
			Power supply fall time	2.90	2.96	3.02	V
		V_{LVD4}	Power supply rise time	2.86	2.92	2.97	V
			Power supply fall time	2.80	2.86	2.91	V
		V_{LVD5}	Power supply rise time	2.76	2.81	2.87	V
			Power supply fall time	2.70	2.75	2.81	V
		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		t_{LD}				300	μs

LVD Detection Voltage of Interrupt & Reset Mode

(T_A = –40 to +85°C, $V_{PDR} \le EV_{DD} = V_{DD} \le 5.5$ V, $V_{SS} = EV_{SS} = 0$ V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVDA0}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 0, 0, falling reset voltage			1.60	1.63	1.66	V
	V_{LVDA1}		LVIS1, LVIS0 = 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}		LVIS1, LVIS0 = 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}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	2.86	2.92	2.97	V
				Falling interrupt voltage	2.80	2.86	2.91	V
	V_{LVDB1}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 0, 1, falling reset voltage			1.80	1.84	1.87	V
	V_{LVDB2}		LVIS1, LVIS0 = 1, 0	Rising release reset voltage	1.94	1.98	2.02	V
				Falling interrupt voltage	1.90	1.94	1.98	V
	V_{LVDB3}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.05	2.09	2.13	V
				Falling interrupt voltage	2.00	2.04	2.08	V
	V_{LVDB4}		LVIS1, LVIS0 = 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_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 0, falling reset voltage			2.40	2.45	2.50	V
	V_{LVDC1}		LVIS1, LVIS0 = 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}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.66	2.71	2.76	V
				Falling interrupt voltage	2.60	2.65	2.70	V
	V_{LVDC3}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.68	3.75	3.82	V
				Falling interrupt voltage	3.60	3.67	3.74	V
	V_{LVDD0}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 1, falling reset voltage			2.70	2.75	2.81	V
	V_{LVDD1}		LVIS1, LVIS0 = 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}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.96	3.02	3.08	V
				Falling interrupt voltage	2.90	2.96	3.02	V
	V_{LVDD3}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.98	4.06	4.14	V
				Falling interrupt voltage	3.90	3.98	4.06	V

27.6.5 Supply voltage rise time

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

Caution Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 27.4 AC Characteristics.

27.7 LCD Characteristics

27.7.1 Resistance division method

(1) Static display mode

(T_A = –40 to +85°C, V_{L4} (MIN.) ≤ V_{DD} ≤ 5.5 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} ≤ 5.5 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} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
LCD drive voltage	V_{L4}		2.5		V_{DD}	V

27.7.2 Internal voltage boosting method

(1) 1/3 bias method

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 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	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_{L4}	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] = 0.47 μF		500			ms

Notes 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%

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).

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} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
LCD output voltage variation range	V_{L1} Note 4	C1 to C5 Note 1 = 0.47 μF	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 C5 Note 1 = 0.47 μF		2 V_{L1} – 0.08	2 V_{L1}	2 V_{L1}	V
Tripler output voltage	V_{L3}	C1 to C5 Note 1 = 0.47 μF		3 V_{L1} – 0.12	3 V_{L1}	3 V_{L1}	V
Quadruply output voltage	V_{L4} Note 4	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 = 0.47 μF		500			ms

Notes 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 = C5 = 0.47 μF±30%

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).

3. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

4. V_{L4} must be 5.5 V or lower.

27.7.3 Capacitor split method

1/3 bias method

(T_A = –40 to +85°C, 2.2 V ≤ V_{DD} ≤ 5.5 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]	t_{VWAIT}		100			ms

Notes 1. This is the wait time from when voltage bucking is started (VLCON = 1) until display is enabled (LCDON = 1).

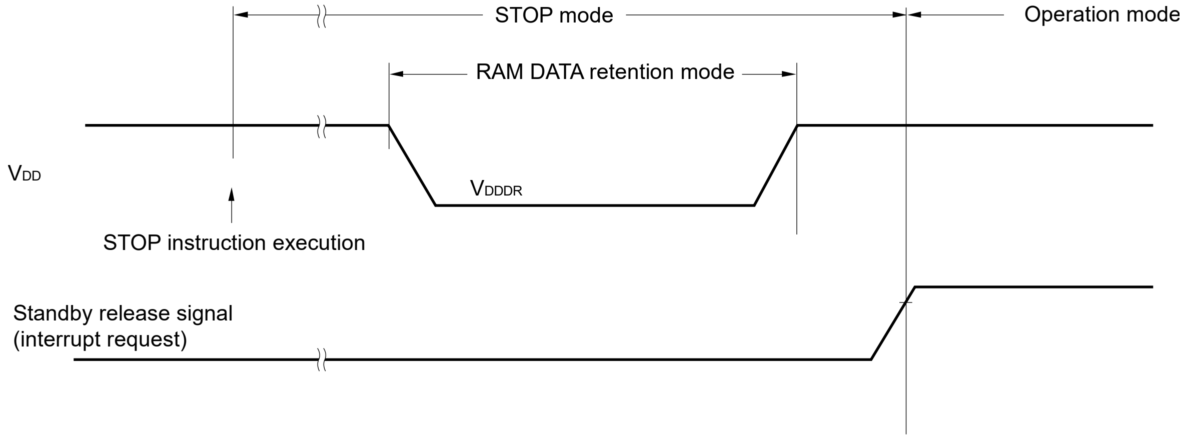
2. This is a capacitor that is connected between voltage pins used to drive the LCD.
C1: A capacitor connected between CAPH and CAPL
C2: A capacitor connected between V_{L1} and GND
C3: A capacitor connected between V_{L2} and GND
C4: A capacitor connected between V_{L4} and GND
C1 = C2 = C3 = C4 = 0.47 μF±30%

27.8 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}		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the data in RAM are retained until a POR is applied, but are not retained following a POR.



27.9 Flash Memory Programming Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency	f_{CLK}	1.8 V ≤ V_{DD} ≤ 5.5 V	1		24	MHz
Number of code flash rewrites **Notes 1, 2, 3**	C_{erwr}	Retained for 20 years **Note 4** T_A = 85°C	1,000			Times
Number of data flash rewrites **Notes 1, 2, 3**		Retained for 1 year **Note 4** T_A = 25°C		1,000,000		
		Retained for 5 years **Note 4** T_A = 85°C	100,000			
		Retained for 20 years **Note 4** T_A = 85°C	10,000			

Notes 1. 1 erase + 1 write after the erase is regarded as 1 rewrite.
The retaining years are until next rewrite after the rewrite.

2. When using flash memory programmer and Renesas Electronics self programming library

3. This characteristic indicates the flash memory characteristic and based on Renesas Electronics reliability test.

4. This temperature is the average value at which data are retained.

Remark When updating data multiple times, use the flash memory as one for updating data.

27.10 Dedicated Flash Memory Programmer Communication (UART)

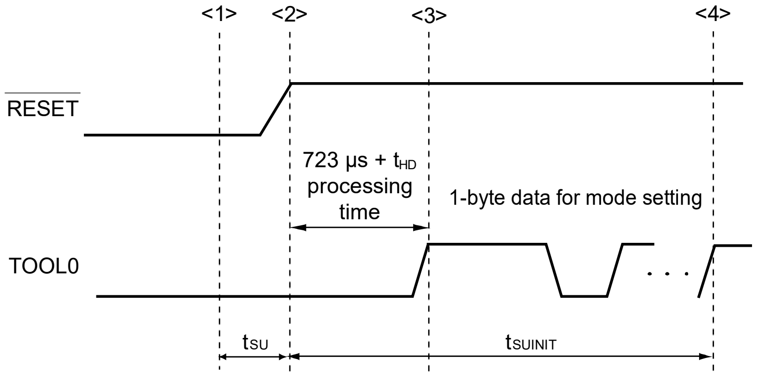
(T_A = –40 to +85°C, 1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During flash memory programming	115,200		1,000,000	bps

27.11 Timing Specifications for Switching Flash Memory Programming Modes

(T_A = –40 to +85°C, 1.8 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Time to complete the communication for the initial setting after the external reset is released	t_{SUINIT}	POR and LVD reset must be released before the external reset is released.			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	t_{SU}	POR and LVD reset must be released before the external reset is released.	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset must be released before the external reset is released.	1			ms



<1> The low level is input to the TOOL0 pin.

<2> The external reset is released (POR and LVD reset must be released before the external reset is released.).

<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}: Communication for the initial setting must be completed within 100 ms after a reset is released during this period.

t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level

t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released (excluding the processing time of the firmware to control the flash memory)

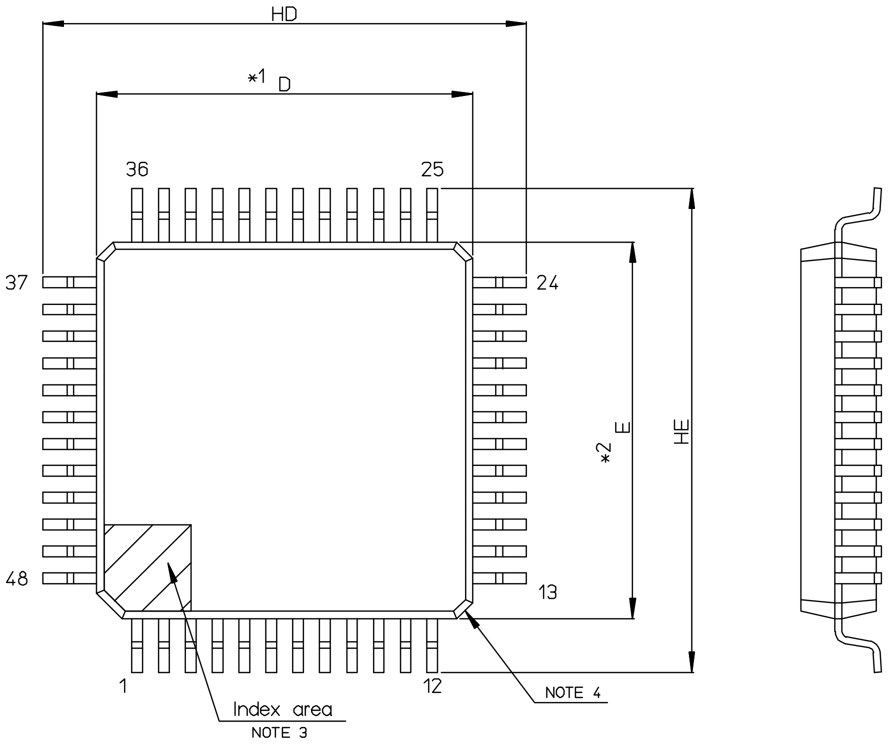
CHAPTER 28 PACKAGE DRAWINGS

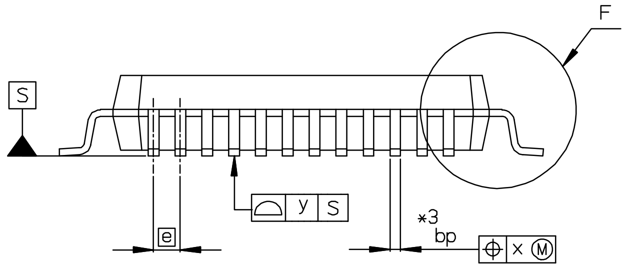
28.1 48-pin Products

R7F0C001G2DFB, R7F0C002G2DFB

<R>

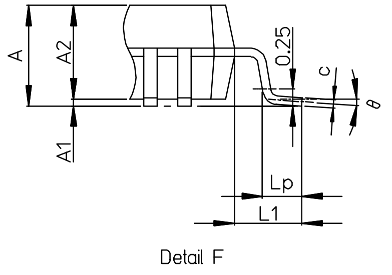
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP48-7x7-0.50	PLQP0048KB-B	——	0.2g





NOTE)

1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL; SIZE MAY VARY.

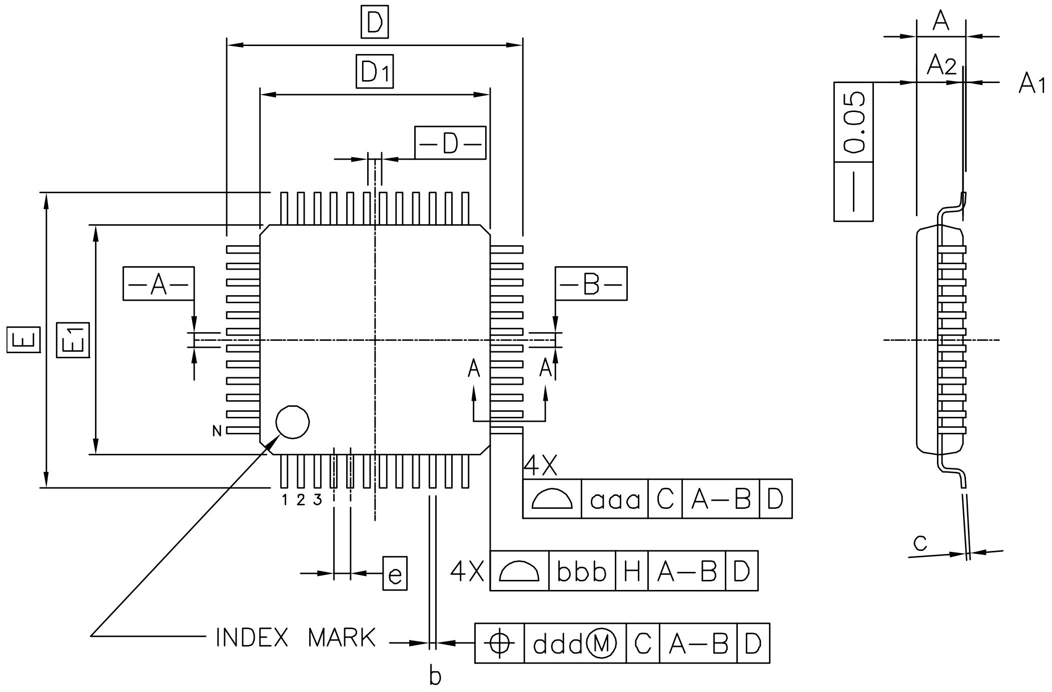


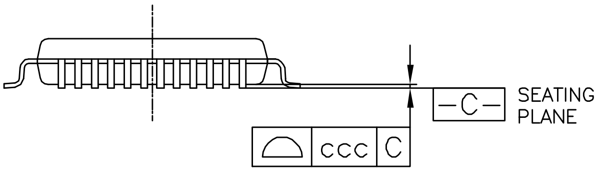
Detail F

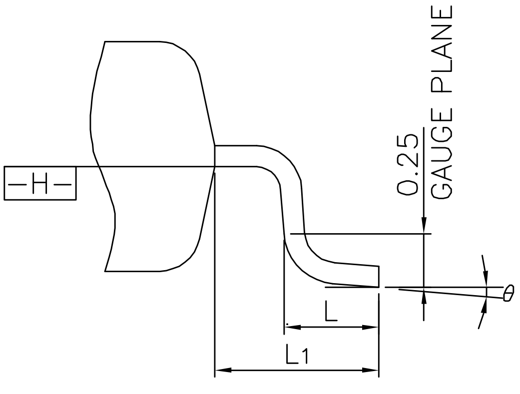
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A2	——	1.4	——
HD	8.8	9.0	9.2
HE	8.8	9.0	9.2
A	——	——	1.7
A1	0.05	——	0.15
bp	0.17	0.20	0.27
c	0.09	——	0.20
θ	0°	3.5°	8°
e	——	0.5	——
x	——	——	0.08
y	——	——	0.08
Lp	0.45	0.6	0.75
L1	——	1.0	——

<R>

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFQFP48-7x7-0.50	PLQP0048KL-A	0.18







SECTION A—A

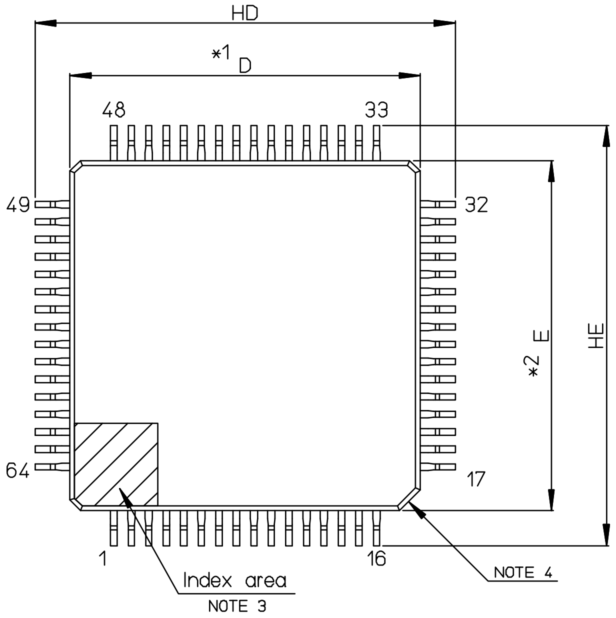
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	1.60
A_1	0.05	—	0.15
A_2	1.35	1.40	1.45
D	—	9.00	—
D_1	—	7.00	—
E	—	9.00	—
E_1	—	7.00	—
N	—	48	—
e	—	0.50	—
b	0.17	0.22	0.27
c	0.09	—	0.20
θ	0°	3.5°	7°
L	0.45	0.60	0.75
L_1	—	1.00	—
aaa	—	—	0.20
bbb	—	—	0.20
ccc	—	—	0.08
ddd	—	—	0.08

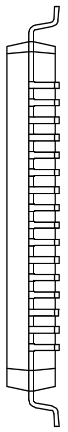
28.2 64-pin Products

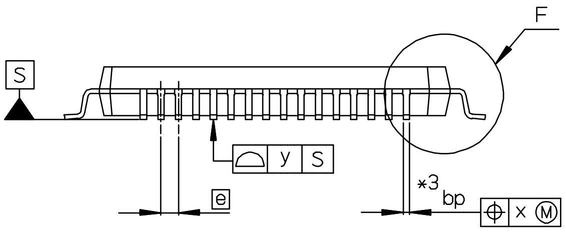
R7F0C001L2DFB, R7F0C002L2DFB

<R>

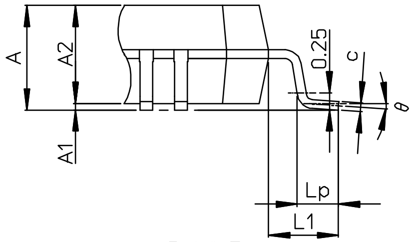
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	—	0.3g







NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL; SIZE MAY VARY.

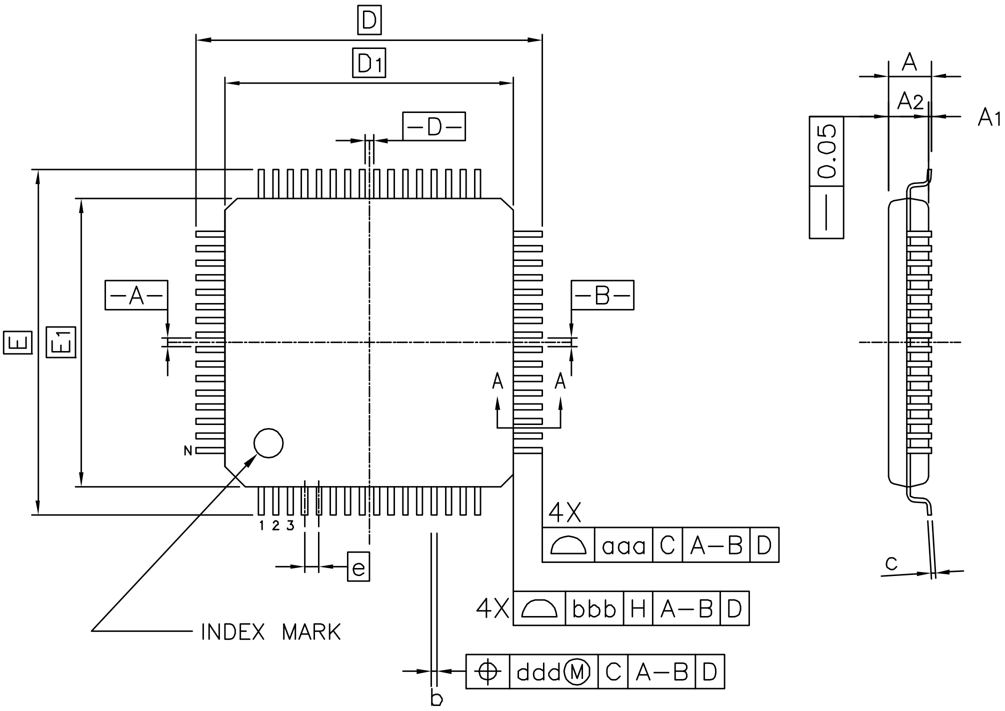


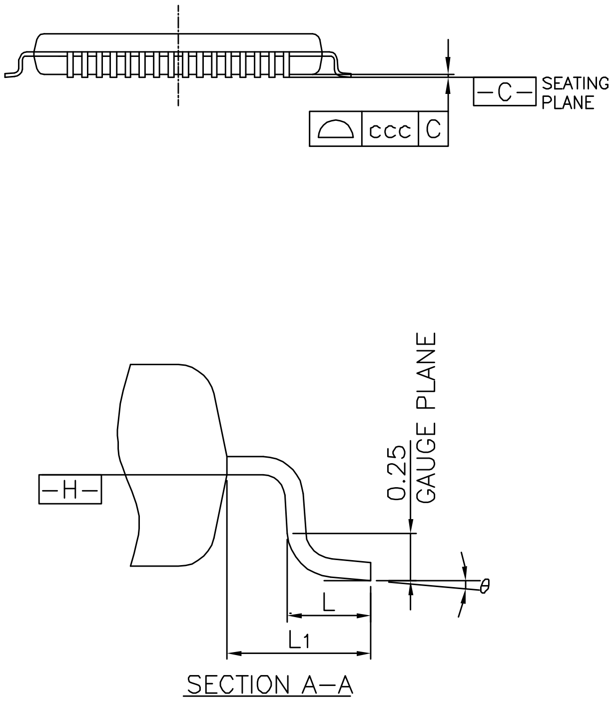
Detail F

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A2	—	1.4	—
HD	11.8	12.0	12.2
HE	11.8	12.0	12.2
A	—	—	1.7
A1	0.05	—	0.15
bp	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
$\boxed{e}$	—	0.5	—
x	—	—	0.08
y	—	—	0.08
Lp	0.45	0.6	0.75
L1	—	1.0	—

<R>

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFQFP064-10x10-0.50	PLQP0064KL-A	0.36





SECTION A-A

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	–	12.00	–
D_1	–	10.00	–
E	–	12.00	–
E_1	–	10.00	–
N	–	64	–
e	–	0.50	–
b	0.17	0.22	0.27
c	0.09	–	0.20
θ	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
ALL		
-	Modification of 3-wire serial I/O, 3-wire serial to Simplified SPI	(c)
-	Modification of CSI to Simplified SPI (CSI)	(c)
-	Deletion overline of SCK	(a)
CHAPTER 1 OUTLINE		
p.2	Addition of Notes in **1.1 Features**	(c)
p.2	Modification of Notes 2 to Notes 3	(c)
p.3	Modification of **Figure 1-1 Part Number, Memory Size, and Package of R7F0C001G/L, R7F0C002G/L**	(d)
p.3	Modification of **Table 1-1. List of Ordering Part Numbers**	(d)
CHAPTER 4 PORT FUNCTIONS		
p.100	Addition of Note in **4.4.5 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**	(c)
p.105	Modification of **Table 4-6. Setting Examples of Registers and Output Latches When Using Alternate Function (1/6)**	(a)
CHAPTER 7 REAL-TIME CLOCK		
p.271	Modification of **Figure 7-5. Format of Real-time Clock Control Register 1 (RTCC1) (2/2)**	(c)
p.283	Modification of Caution in **Figure 7-21. Procedure for Reading Real-time Clock**	(c)
p.285	Modification of Caution in **Figure 7-22. Procedure for Writing Real-time Clock**	(c)
CHAPTER 12 SERIAL ARRAY UNIT		
p.365	Addition of Note in **CHAPTER 12 SERIAL ARRAY UNIT**	(c)
CHAPTER 20 SAFETY FUNCTIONS		
p.625	Modification of **20.1 Overview of Safety Functions**	(c)
p.630	Modification of **20.3.2 CRC operation function (general-purpose CRC)**	(c)
p.634	Modification of **20.3.4 RAM guard function**	(c)
p.635	Modification of **20.3.5 SFR guard function**	(c)
CHAPTER 23 FLASH MEMORY		
p.676	Addition of Cautions 4 in **23.8.3 Procedure for accessing data flash memory**	(c)
CHAPTER 27 ELECTRICAL SPECIFICATIONS		
p.716	Modification of Notes 1 and 4 in **27.3.2 Supply current characteristics**	(c)
p.718	Modification of Notes 1 and 5 in **27.3.2 Supply current characteristics**	(c)
p.718	Deletion of Notes 6 in **27.3.2 Supply current characteristics**	(c)
CHAPTER 28 PACKAGE DRAWINGS		
p.759	Deletion of PLQP0048KF-A in **28.1 48-pin Products**	(d)
p.759	Addition of PLQP0048KB-B in **28.1 48-pin Products**	(d)
p.760	Addition of PLQP0048KL-A in **28.1 48-pin Products**	(d)
p.761	Deletion of PLQP0064KF-A in **28.2 64-pin Products**	(d)
p.761	Addition of PLQP0064KB-C in **28.2 64-pin Products**	(d)
p.762	Addition of PLQP0064KL-A in **28.2 64-pin Products**	(d)

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.

(1/17)

Edition	Description	Chapter
Rev.2.20	Modification of figure in **1.3.1 48-pin products**	**CHAPTER 1 OUTLINE**
	Modification of figure in **1.3.2 64-pin products**	
	Modification of table in **1.6 Outline of Functions**	
	Modification of table in **2.1.1 48-pin products**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of table in **2.1.2 64-pin products**	
	Addition of **caution** in **Figure 2-8 Pin Block Diagram for Pin Type 7-5-7**	
	Addition of **caution** and modification of figure in **Figure 2-10 Pin Block Diagram for Pin Type 8-5-1**	
	Addition of **caution** in **Figure 2-11 Pin Block Diagram for Pin Type 8-5-2**	
	Addition of **caution** in **Figure 2-12 Pin Block Diagram for Pin Type 8-5-3**	
	Addition of **cautions1** and **2** in **Figure 2-13 Pin Block Diagram for Pin Type 8-5-7**	
	Addition of **caution** in **Figure 2-14 Pin Block Diagram for Pin Type 12-1-4**	
	Modification of **Table 3-3 Vector Table**	**CHAPTER 3 CPU ARCHITECTURE**
	Modification of description in **3.2.1 Control registers (1) Program counter (PC)**	
	Addition of description in **5.1 Functions of Clock Generator (1) Main system clock**	**CHAPTER 5 CLOCK GENERATOR**
	Modification of **caution 6** in **Figure 5-4 Format of Clock Operation Status Control Register (CSC)**	
	Modification of description in **5.4.4 Low-speed on-chip oscillator**	
	Addition of **caution** in **5.6.2 Example of setting X1 oscillation clock**	
	Modification of description in **5.6.3 Example of setting XT1 oscillation clock**	
	Modification of **Table 5-4 Changing CPU Clock (1/2)**	
	Addition of description in 5.6.7 Conditions before clock oscillation is stopped	
	Modification of description in **5.7 Resonator and Oscillator Constants**	
	Modification of table in **5.7 Resonator and Oscillator Constants (2) XT1 oscillation: Crystal resonator**	
	Modification of **Figure 6-11 Format of Timer Mode Register mn (TMRmn) (1/4)**	**CHAPTER 6 TIMER ARRAY UNIT**
	Modification of **Figure 6-11 Format of Timer Mode Register mn (TMRmn) (4/4)**	
	Modification of **caution** in **Figure 6-16 Format of Timer Input Select register 0 (TIS0)**	
	Modification of **Figure 6-22 Format of Noise Filter Enable Register 1 (NFEN1)**	
	Modification of description in **6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only)**	
	Deletion of caution in Figure 6-38 TO0n Pin Statuses by Collective Manipulation of TO0n Bit	
	Modification of description in **6.8.2 Operation as external event counter**	
	Modification of **Figure 6-62 Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used**	
	Modification of description in **6.9.1 Operation as one-shot pulse output function**	
	Addition of **notes 1** and **2** in **Figure 7-5 Format of Real-time Clock Control Register 1 (RTCC1)**	**CHAPTER 7 REAL-TIME CLOCK**

(2/17)

Edition	Description	Chapter
Rev.2.20	Modification of description in **9.5 Cautions of clock output/buzzer output controller**	**CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER**
	Addition of **note** in **Table 10-3 Setting of Overflow Time of Watchdog Timer**	**CHAPTER 10 WATCHDOG TIMER**
	Addition of **note** in **Table 10-4 Setting Window Open Period of Watchdog Timer**	
	Modification of **Table 10-5 Setting of Watchdog Timer Interval Interrupt**	
	Addition of **Figure 11-32 Flowchart for Setting up SNOOZE Mode**	**CHAPTER 11 A/D CONVERTER**
	Modification of description in **11.10 Cautions for A/D Converter (2) Input range of ANI0, ANI1 and ANI16 to ANI23 pins**	
	Modification of **note** in **12.2.2 Lower 9 bits of the serial data register mn (SDRmn)**	**CHAPTER 12 SERIAL ARRAY UNIT**
	Modification of **caution 2** in **Figure 12-7 Format of Serial Data Register mn (SDRmn)**	
	Modification of description and **remark** in **Figure 12-16 Examples of Reverse Transmit Data**	
	Modification of description in **12.5.7 SNOOZE mode function**	
	Modification of description and **note** in **Figure 12-68 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)**	
	Modification of **Figure 12-69 Flowchart of SNOOZE Mode Operation (once startup)**	
	Modification of Figure and **note** in **Figure 12-70 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)**	
	Modification of **Figure 12-71 Flowchart of SNOOZE Mode Operation (continuous startup)**	
	Modification of table in **12.6.2 UART reception**	
	Modification of **Figure 12-84 Procedure for Resuming UART Reception Stop**	
	Modification of description and **caution 5** in **12.6.3 SNOOZE mode function**	
	Modification of **Figure 12-87 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)**	
	Modification of **Figure 12-88 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)**	
	Modification of **Figure 12-89 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**	
	Modification of **Figure 12-90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**	
	Modification of **Figure 12-91 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**	
	Modification of **Figure 13-26 Voltages and Phases of Common and Segment Signals (3/3) (e) 1/4 bias method (waveform A)**	**CHAPTER 13 LCD CONTROLLER/DRIVER**
	Modification of **Table 14-1 Interrupt Source List**	**CHAPTER 14 INTERRUPT FUNCTIONS**
	Modification of table and **note 3** in **Table 14-1 Interrupt Source List**	
	Modification of **Figure 14-3 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L)(64-pin products)**	
	Modification of **Table 14-5 Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing**	

(3/17)

Edition	Description	Chapter
Rev.2.00	Modification of description in **CHAPTER 17 RESET FUNCTION**	**CHAPTER 17 RESET FUNCTION**
	Addition of description in **17.1 Timing of Reset Operation**	
	Modification of figure title and deletion of caution in Figure 17-3 Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal Memory Access	
	Modification of **Table 17-2 State of Hardware After Receiving a Reset Signal**	
	Modification of **notes 3** and **4** in **Figure 18-2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3)**	**CHAPTER 18 POWER-ON-RESET CIRCUIT**
	Modification of table in **19.1 Functions of Voltage Detector**	**CHAPTER 19 VOLTAGE DETECTOR**
	Modification of description in **19.4.2 When used as interrupt mode**	
	Modification of **19.4.3 When used as interrupt and reset mode**	
	Addition of description in **22.1.1 User option byte (000C0H to 000C2H) (3)**	**CHAPTER 22 OPTION BYTE**
	Addition of **note 2** in **Figure 22-1 Format of User Option Byte (000C0H)**	
	Modification of **Figure 22-3. Format of Option Byte (000C2H)**	
	Modification of **Table 23-1 Wiring Between R7F0C001G/L, R7F0C002G/L and Dedicated Flash Memory Programmer**	**CHAPTER 23 FLASH MEMORY**
	Modification of **Figure 23-2 Communication with Dedicated Flash Memory Programmer**	
	Modification of **Table 23-2 Pin Connection**	
	Modification of **Table 24-1 On-Chip Debug Security ID**	**CHAPTER 24 ON-CHIP DEBUG FUNCTION**
	Modification of **27.8 RAM Data Retention Characteristics**	**CHAPTER 27 ELECTRICAL SPECIFICATIONS**
	Modification of table and addition of **note 4** to **27.9 Flash Memory Programming Characteristics**	

(4/17)

Edition	Description	Chapter
Rev.2.10	Modification of description in **27.2.2 On-chip oscillator characteristics**	**CHAPTER 27 ELECTRICAL SPECIFICATIONS**
Rev.2.00	Renamed operation speed mode control register to subsystem clock supply mode control register (OSMC)	Though out
	Modification of **1.1 Features**	**CHAPTER 1 OUTLINE**
	Modification of **1.2 List of Part Numbers**	
	Modification of description in **1.4 Pin Identification**	
	Modification of tables in **2. 1. 1** to **2. 1. 2**	**CHAPTER 2 PIN FUNCTIONS**
	Addition of **2.2.2 Description of Functions**	
	Modification of **Table 2-3. Connection of Unused Pins**	
	Addition of **Figures 2-1** to **2-14** in **2.4 Block Diagrams of Pins**	
	Modification of **cautions 2** and **3** in **3.1.3 Internal data memory space**	**CHAPTER 3 CPU ARCHITECTURE**
	Figures 3-4 to **3-5** in **3.1.6 Data memory addressing** were consolidated into **Figure 3-4**	
	Modification of description in **Table 3-5**.	
	Modification of description in **Table 3-6**	
	Addition of description of port 1 to port 14 in **4.2.1** to **4.2.10**	**CHAPTER 4 PORT FUNCTIONS**
	Addition of **caution** to **Figure 4-1**	
	Addition of **caution** to **Figure 4-2**	
	Addition of **caution 2** to **Figure 4-3**	
	Addition of **caution 2** to **Figure 4-4**	
	Addition of **caution** in **4.3.5**	
	Addition of **caution** to **Figure 4-5**	
	Modification of **caution 1** in **Figure 4-6**	
	Addition of **caution 3** to **Figure 4-6**	
	Addition of **caution 3** to **Figure 4-7**	
	Modification of description in **4.4.4 Connecting to external device with different potential (1.8 V, 2.5 V, 3 V)**	
	Addition of **4.4.5 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**	
	Addition of **4.5.1 Basic concept when using alternate function**	
	Addition of **4.5.2 Register settings for alternate function whose output function is not used**	
	Addition of **4.5.3 Register setting examples for used port and alternate functions**	
	Modification of description in **4.6.2 Notes on specifying the pin settings**	
	Addition of description of port 1 to port 14 in **4.2.1** to **4.2.10**	
	Modification of description in **5.3.5 Oscillation stabilization time select register (OSTS)**	**CHAPTER 5 CLOCK GENERATOR**
	Modification of **cautions 2** and **3** in **Figure 5-6**	
	Modification of description in **Figure 5-8**	
	Modification of description 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 **5.6.1 Example of setting high-speed on-chip oscillator**	
	Modification of description in **5.6.3 Example of setting XT1 oscillation clock**	

(5/17)

Edition	Description	Chapter
Rev.2.00	Modification of description in **Figure 5-14**	**CHAPTER 5 CLOCK GENERATOR**
	Modification of **caution** in **Table 5-3 (2/5)**	
	Modification of description in **Table 5-3 (3/5)**	
	Addition of **remark 2** to **Table 5-3 (3/5)**	
	Modification of description in **Table 5-3 (4/5)**	
	Addition of **5.7 Resonator and Oscillator Constants**	
	Modification of description of the timer array unit	**CHAPTER 6 TIMER ARRAY UNIT**
	Modification of description in **6.1.2 Simultaneous channel operation function (1) One-shot pulse output**	
	Modification of description in **Figure 6-1**	
	Modification of description in **Figures 6-2** and **6-3**	
	Modification of description in **Figures 6-4** and **6-5**	
	Modification of **caution 1** in **6.3.1 Peripheral enable register 0 (PER0)**	
	Modification of description in **Figure 6-18**	
	Modification of description in **6.3.14 Noise filter enable register 1 (NFEN1)**	
	Modification of **remark** in **6.5.3 Operation of counter (2) Operation of event counter mode**	
	Modification of description and **remark** in **Figure 6-28**	
	Modification of description in **6.7 Timer Input (TImn) Control**	
	Modification of **Figure 6-47**	
	Modification of description in **6.8.4 Operation as input pulse interval measurement**	
	Modification of **Figure 6-55**	
	Modification of description in **Figure 6-58**	
	Modification in **Figure 6-59**	
	Modification of description in **Figure 6-62**	
	Modification of **Figure 6-63**	
	Modification of description in **Figure 6-66**	
	Modification of **Figure 6-67**	
	Modification of description in **Figure 6-71**	
	Modification of description in **7.1 Functions of Real-time Clock**	**CHAPTER 7 REAL-TIME CLOCK**
	Modification of description in **7.3 Registers Controlling Real-time Clock**	
	Modification of **cautions 1** and **2** in **Figure 7-2**	
	Addition of **caution 2** in **Figure 7-4**	
	Addition of **caution 2** in **Figure 7-6**	
	Addition of description in **7.3.16 Port mode register 3 (PM3)**	
	Addition of description in **7.3.17 Port register 3 (P3)**	
	Modification of **note 1** in **Figure 7-19**	
	Addition of **correction example 1** and **Figure 7-25**	
	Modification of **cautions 1** and **2** in **Figure 8-2**	**CHAPTER 8 12-BIT INTERVAL TIMER**
	Addition of description in **8.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode**	

(6/17)

Edition	Description	Chapter
Rev.2.00	Modification of **caution** in **9.1 Functions of Clock Output/Buzzer Output Controller**	**CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER**
	Modification of **cautions 2** and **3** in **Figure 9-3**	
	Modification of description in **9.4.1 Operation as output pin**	
	Addition of description in **10.1 Functions of Watchdog Timer**	**CHAPTER 10 WATCHDOG TIMER**
	Modification of description in **Table 10-1**	
	Modification of description in **Figure 10-1**	
	Modification of **caution 2** in **10.4.1 Controlling operation of watchdog timer**	
	Modification of description in **11.1 Function of A/D Converter**	**CHAPTER 11 A/D CONVERTER**
	Modification of description in **Figure 11-1**	
	Modification of description in **11.2 Configuration of A/D Converter**	
	Modification of **caution 4** in **Figure 11-4**	
	Addition of **note 3** and **caution 1** in **Table 11-3 (1/4)**	
	Addition of **note 8** and **caution 1** in **Table 11-3 (2/4)**	
	Addition of **note 4** and **caution 1** in **Table 11-3 (3/4)**	
	Addition of **note 9** and **caution 1** in **Table 11-3 (4/4)**	
	Modification of **Figure 11-5**	
	Modification of **cautions 2** and **3** in **Figure 11-6**	
	Modification of description of ADREFP1, ADREFP0, ADRCK, and AWC bits in **Figure 11-7**	
	Modification of **caution 8** in **Figure 11-11**	
	Addition of **cautions 2** and **3** in **Figure 11-13**	
	Modification of description in **11.3.10 A/D test register (ADTES)**	
	Modification of description of ADTES1 and ADTES0 bits in **Figure 11-14**	
	Modification of description in **11.3.11 Registers controlling port function of analog input pins**	
	Modification of **Figure 11-15** and description	
	Modification of **Figure 11-20**	
	Modification of **Figure 11-22**	
	Modification of **Figure 11-23**	
	Modification of **Figure 11-24**	
	Modification of **Figure 11-25**	
	Modification of **Figure 11-26**	
	Modification of **Figure 11-27** and **caution**	
	Modification of description in **11.8 SNOOZE Mode Function**	
	Modification of **Figures 11-34** and **11-35**	
	Modification of description in **11.10 (5) Analog input (ANIn) pins**	
	Modification of description in **11.10 (6) Input impedance of analog input (ANIn) pins**	
	Modification of description in **12.1.1 3-wire serial I/O (CSI00, CSI01)**	**CHAPTER 12 SERIAL ARRAY UNIT**
	Modification of **Figure 12-1**	
	Modification of **caution 1** in **Figure 12-3**	

(7/17)

Edition	Description	Chapter
Rev.2.00	Modification of description in **12.3.5 Higher 7 bits of the serial data register mn (SDRmn)**	**CHAPTER 12 SERIAL ARRAY UNIT**
	Addition of **Figure 12-16**	
	Modification of **caution** in **12.3.14 Serial standby control register m (SSCm)**	
	Modification of description in **Figure 12-17**	
	Addition of **Figure 12-18**	
	Addition of **12.3.16 Registers controlling port functions of serial input/output pins**	
	Modification of description in **12.5 Operation of 3-Wire Serial I/O (CSI00, CSI01) Communication**	
	Modification of **Figure 12-25**	
	Modification of **Figure 12-27**	
	Modification of **Figure 12-29**	
	Modification of **Figure 12-32**	
	Modification of **Figure 12-33**	
	Modification of **Figure 12-35**	
	Modification of **Figure 12-37**	
	Modification of **Figure 12-43**	
	Modification of **Figure 12-45**	
	Modification of **note 1** in **12.5.4 Slave transmission**	
	Modification of **Figure 12-49**	
	Modification of **Figure 12-51**	
	Modification of **Figure 12-53**	
	Modification of description and **note 1** in **12.5.5 Slave reception**	
	Modification of **Figure 12-59**	
	Modification of **note 1** in **12.5.6 Slave transmission/reception**	
	Addition of **caution 2** in **Figure 12-63**	
	Modification of **Figure 12-65**	
	Modification of **Figure 12-67**	
	Modification of description and **caution 2** in **12.5.7 SNOOZE mode function**	
	Modification of **caution 2** in **Figure 12-70**	
	Modification of **Figure 12-75**	
	Modification of **Figure 12-78**	
	Modification of **Figure 12-80**	
	Modification of **Figure 12-86**	
	Modification of description in **12.6.3 SNOOZE mode function**	
	Addition of **Table 12-3**	
	Modification of description in **(1) SNOOZE mode operation (EOCm1 = 0, SSECm = 0/1)** in **12.6.3 SNOOZE mode function**	
	Modification of description in **(2) SNOOZE mode operation (EOCm1 = 1, SSECm = 0: Error interrupt (INTSREq) generation is enabled)** in **12.6.3 SNOOZE mode function**	
	Modification of **Figure 12-89**	
	Modification of description in **(3) SNOOZE mode operation (EOCm1 = 1, SSECm = 1: Error interrupt (INTSREq) generation is stopped)** in **12.6.3 SNOOZE mode function**	
	Modification of **Figure 12-91**	

(8/17)

Edition	Description	Chapter
Rev.2.00	Modification of **Tables 13-1**	**CHAPTER 13 LCD CONTROLLER/DRIVER**
	Modification of **Figure 13-1**	
	Modification of **caution 1** in **Figure 13-2**	
	Modification of **caution 1** in **Figure 13-4 (2/2)**	
	Modification of **cautions 1** and **3** in **Figure 13-6 (2/2)**	
	Modification of **Figure 13-20** and **caution**	
	Modification of **note 3** in **Table 14-2 (2/3)**	**CHAPTER 14 INTERRUPT FUNCTION**
	Modification of **caution 1** in **Figure 14-2**	
	Modification of description in **14.3.5 Program status word (PSW)**	
	Modification of **Figure 14-9**	
	Modification of description and **remark 3** in **Table 14-5**	
	Modification of description in **Table 15-1**	**CHAPTER 15 KEY INTERRUPT FUNCTION**
	Modification of description in **Table 15-2**	
	Addition of **caution** in **16.3.1 HALT mode**	**CHAPTER 16 STANDBY FUNCTION**
	Modification of **note 2** in **Figure 16-1**	
	Modification of **Figures 16-2 (1)** and **(2)**	
	Modification of **Figure 16-3 (1), note 2,** and **remark 1**	
	Modification of **Figure 16-3 (2)** and **note 2**	
	Addition of **caution** in **16.3.1 HALT mode**	
	Modification of **note 2**, **caution**, and **remarks** in **Figure 16-3 (3)**	
	Modification of **Figures 16-4 (1)** and **(2)**	
	Modification of description in **16.3.3 SNOOZE mode**	
	Modification of description, **caution 3**, and **remark**	**CHAPTER 17 RESET FUNCTION**
	Addition of **note 3** in **Figure 17-3**	
	Modification of description in **Table 17-1** and addition of **note**	
	Modification of **Table 17-2** and **remark**	
	Addition of description in **Table 17-3**	
	Addition of **Figure 17-5**	
	Modification of description, **caution**, and **remarks 1** and **2** in **18.1 Functions of Power-on-reset Circuit**	**CHAPTER 18 POWER-ON-RESET CIRCUIT**
	Modification description in **18.3 Operation of Power-on-reset Circuit**	
	Modification of description in **Figure 18-2 (1/3)**	
	Modification of description in **Figure 18-2 (2/3)**	
	Modification of description in **Figure 18-2 (3/3)**	
	Modification of description in **19.1 Functions of Voltage Detector**	**CHAPTER 19 VOLTAGE DETECTOR**
	Modification of description in **Figure 19-2**	
	Modification of **note 2** in **Figure 19-3**	
	Modification of description in **Table 19-1 (1/2)**	
	Modification of description in **Table 19-1 (2/2)**	

(9/17)

Edition	Description	Chapter
Rev.2.00	Modification of description in **19.4.1 When used as reset mode**	**CHAPTER 19 VOLTAGE DETECTOR**
	Modification of **Figure 19-4**	
	Modification of description in **19.4.2 When used as interrupt mode**	
	Modification of **Figure 19-5**	
	Modification of description in **Figure 19-6 (1/2)**	
	Modification of description in **Figure 19-6 (2/2)**	
	Modification of description in **19.5 Cautions for Voltage Detector**	
	Modification of (6), (7), and **remark** in **20.1 Overview of Safety Functions**	**CHAPTER 20 SAFETY FUNCTIONS**
	Modification of description in **Figure 20-3**	
	Modification of **caution** and **remarks** in **Figure 20-7**	
	Addition of **Figure 20-8**	
	Modification of description and **note** in **Figure 20-11**	
	Modification of description in **20.3.7 Frequency detection function**	
	Modification of description in **20.3.8 A/D test function**	
	Modification of description in **20.3.8.1 A/D test register (ADTES)**	
	Modification of description and caution 1 to 9 in **20.3.8.2 Analog input channel specification register (ADS)**	
	Modification of description of (2) and (3) in **22.1.1 User option byte (000C0H to 000C2H)**	**CHAPTER 22 OPTION BYTE**
	Modification of description in **Figure 22-2 (1/2)**	
	Modification of description, **caution 2**, and **remark 3** in **Figure 22-2 (2/2)**	
	Modification of **caution 2** in **Figure 22-3**	
	Modification of description	**CHAPTER 23 FLASH MEMORY**
	Modification of **note** in **Table 23-1**	
	Modification of **note 3** in **Figure 23-2**	
	Modification of description, **note**, and **caution** in **Table 23-2**	
	Modification of **note** in **Figure 23-4**	
	Modification of description and **note** in **Table 23-3**	
	Modification of **remark 1** in **23.3.1 P40/TOOL0 pin**	
	Addition of description in **23.4 Serial Programming Method**	
	Addition of description in **23.5 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)**	
	Addition of description in **23.6 Self-Programming**	
	Modification of description in **23.7 Security Settings**	
	Addition of description in **23.8.1 Data flash overview**	
	Addition of description in **23.8.2 Register controlling data flash memory**	
	Addition of description in **23.8.3 Procedure for accessing data flash memory**	
	Modification of description	
	Addition of addr5 in **Table 26-2**	**CHAPTER 26 INSTRUCTION SET**
	Modification of **note2** in **Table 26-5 (1/17) to (17/17)**	
	Modification of description in **Absolute Maximum Ratings (T_A = 25°C) (1/3)**	**CHAPTER 27 ELECTRICAL SPECIFICATIONS**
	Modification of description and **note 2** in **Absolute Maximum Ratings (T_A = 25°C) (2/3)**	

(10/17)

Edition	Description	Chapter
Rev.2.00	Modification of table, **note**, **caution**, and **remark** in **27.2.1 X1, XT1 oscillator characteristics**	**CHAPTER 27 ELECTRICAL SPECIFICATIONS**
	Modification of table, **note**, **caution**, and **remark** in **27.2.2 On-chip oscillator characteristics**	
	Modification of table, **notes 2** and **3** in **27.3.1 Pin characteristics (1/5)**	
	Modification of **notes 3** in **27.3.1 Pin characteristics (2/5)**	
	Modification of **notes 1** and **4** in **27.3.2 Supply current characteristics (1/3)**	
	Modification of table, **notes 1, 5,** and **6** in **27.3.2 Supply current characteristics (2/3)**	
	Modification of table, **notes 1, 3, 4, 5** and **7 to 10** in **27.3.2 Supply current characteristics (3/3)**	
	Modification of table in **27.4 AC Characteristics**	
	Addition of **Minimum Instruction Execution Time during Main System Clock Operation**	
	Modification of **AC Timing Test Points** and **External System Clock Timing**	
	Modification of **AC Timing Test Points**	
	Modification of description, **notes 1** and **2** in **(1) During communication at same potential (UART mode)**	
	Modification of description, **remark 2** in **(2) During communication at same potential (CSI mode)**	
	Modification of description in **(3) During communication at same potential (CSI mode)**	
	Modification of description, **notes 1** and **3,** and **remark 3** in **(4) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode) (1/2)**	
	Modification of description, and **remark 3** in **(4) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode) (2/2)**	
	Modification of table, and **note 1, caution,** and **remark 3** in **(5) Communication at different potential (2.5 V, 3 V) (CSI mode)**	
	Modification of table and **note** in **(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (1/3)**	
	Modification of table and **notes 1** to **3** in **(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (2/3)**	
	Modification of table, **note 3,** and **remark 3** in **(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (3/3)**	
	Modification of table in **(7) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (1/2)**	
	Modification of table in **(7) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (2/2)**	
	Addition of table in **27.6.1 A/D converter characteristics**	
	Modification of description and **notes 3** to **5** in **27.6.1 (1)**	
	Modification of description, **notes 3** and **4** in **27.6.1 (2)**	
	Modification of description, **notes 3** and **4** in **27.6.1 (3)**	
	Modification of description, **notes 3** and **4** in **27.6.1 (4)**	
	Modification of the table in **27.6.2 Temperature sensor/internal reference voltage characteristics**	
	Modification of the table and **note** in **27.6.3 POR circuit characteristics**	
	Modification of the table of **LVD Detection Voltage of Interrupt & Reset Mode**	

(11/17)

Edition	Description	Chapter
Rev.2.00	Modification from V_{DD} rise slope to Power supply voltage rising slope in **27.6.5 Supply voltage rise time**	**CHAPTER 27 ELECTRICAL SPECIFICATIONS**
	Addition of description in **27.10 Dedicated Flash Memory Programmer Communication (UART)**	
	Modification of the figure in **27.11 Timing Specifications for Switching Flash Memory Programming Modes**	
Rev.1.00	Renamed interval timer (unit) to 12-bit interval timer	Though out
	Renamed V_{LVI}, V_{LVIH}, V_{LVIL} to V_{LVD}, V_{LVDH}, V_{LVDL} (LVD detection voltage)	
	Renamed interrupt source of RAM parity error (RAMTOP) to RPE	
	Modification of description of INTP0 to INTP7 in **1.4 Pin Identification**	**CHAPTER 1 OUTLINE**
	Modification of **1.5 Block Diagram**	
	Addition and Modification of description in **1.6 Outline of Functions**	
	Modification of **2.1 Port Function**	**CHAPTER 2 PIN FUNCTIONS**
	Addition of **remark** to **2.3 Pin I/O Circuits and Recommended Connection of Unused Pins**	
	Modification of **Figures 3-1** and **3-2**	**CHAPTER 3 CPU ARCHITECTURE**
	Addition of **remark** to **Table 3-1. Correspondence Between Address Values and Block Numbers in Flash Memory**	
	Modification of description in **3.1.1 (4) On-chip debug security ID setting area**	
	Modification of description in **3.1.2 Mirror area**	
	Modification of description and **cautions 1, 2** in **3.1.3 Internal data memory space**	
	Modification of **3.2.1 Control registers**, **3.2.2 General-purpose registers**, and **3.2.3 ES and CS registers**	
	Modification of description in **3.2.4 Special function registers (SFRs)**	
	Modification of description in **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**	
	Modification of **Figures 3-12 to 3-14, 3-16 to 3-39**	
	Modification of [Operand format] in **3.4.1 Implied addressing**	
	Modification of [Operand format] in **3.4.3 Direct addressing**	
	Modification of [Function] in **3.4.7 Based addressing**	
	Modification from [Operand format] to [Description format], modification of [Function] and [Description format], and addition of description in **3.4.9 Stack addressing**	
	Modification of description in **4.1 Port Functions**	**CHAPTER 4 PORT FUNCTIONS**
	Addition of **caution** to **4.3 Registers Controlling Port Function**	
	Modification of **Figure 4-2. Format of Port Register (64-pin products)**	
	Modification of description and addition of **caution** to **4.3.3 Pull-up resistor option registers (PUxx)**	
	Addition of description in **4.3.5 Port output mode registers (POM1)**	
	Addition of **cautions 1** and **2** to **Figure 4-6. Format of Port Mode Control Register (64-pin products)**	
	Modification of description in **4.3.8 Peripheral I/O redirection register (PIOR)**	
	Modification of description in **4.4.1 (2) Input mode** and **4.4.3 (2) Input mode**	
	Modification of description in **4.4.4 Connecting to external device with different potential (1.8 V, 2.5 V, 3 V)**	

(12/17)

Edition	Description	Chapter
Rev.1.00	Addition of **caution** to **4.5 Settings of Port Related Register When Using Alternate Function**	**CHAPTER 4 PORT FUNCTIONS**
	Addition of **4.6.2 Notes on specifying the pin settings**	
	Addition of description to **5.1 (1) <2> High-speed on-chip oscillator**	**CHAPTER 5 CLOCK GENERATOR**
	Modification of **cautions 1, 7** and addition of **cautions 4 to 6** to **Figure 5-2. Format of Clock Operation Mode Control Register (CMC)**	
	Deletion of **cautions 1 to 4** and addition of **cautions 1 to 3** to **5.3.8 High-speed on-chip oscillator frequency select register (HOCODIV)**	
	Modification of **caution** in **Figure 5-11. Example of External Circuit of XT1 Oscillator**	
	Modification of **note 3** in **Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On**	
	Modification of description of [Option byte setting] 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**	
	Addition of description to **Figure 5-14. CPU Clock Status Transition Diagram**	
	Addition of description to **Table 5-3. CPU Clock Transition and SFR Register Setting Examples**	
	Modification and deletion of description in **Table 5-4. Changing CPU Clock**	
	Modification of **remark 2** in **5.6.6 Time required for switchover of CPU clock and system clock**	
	Modification of description in **6.1.1 (7) Delay counter**	**CHAPTER 6 TIMER ARRAY UNIT**
	Modification of **caution** in **6.1.2 (4) Remote control output function**	
	Modification of **Figure 6-2. Internal Block Diagram of Channels 0, 2, 6 of Timer Array Unit 0**	
	Addition of **Figures 6-3** to **6-5**	
	Modification of **Table 6-3. Timer Count Register mn (TCRmn) Read Value in Various Operation Modes**	
	Modification of description in **6.2.2 Timer data register mn (TDRmn)**	
	Modification of **note** and **remark 2** and addition of **caution** to **Figure 6-10. Format of Timer Clock Select register m (TPSm) (1/2)**	
	Addition of **note** to **Figure 6-11. Format of Timer Mode Register mn (TMRmn)**	
	Modification of Setting of starting counting and interrupt and **note 3** and addition of **note 1** to **Figure 6-11. Format of Timer Mode Register mn (TMRmn) (4/4)**	
	Modification of description in **Figure 6-15. Format of Timer Channel Stop register m (TTm)**	
	Addition of **caution** to **Figure 6-16. Format of Timer Input Select register 0 (TIS0)**	
	Modification of description in **Figure 6-18. Format of Timer Output Enable register m (TOE0)**	
	Modification of description in **Table 6-6. Operations from Count Operation Enabled State to Timer count Register mn (TCRmn) Count Start**	
	Addition of title to **6.5.3 Operation of counter**	
	Modification of description, **remark** and addition of **note** to **Figure 6-28. Operation Timing (Capture Mode: Input Pulse Interval Measurement)**	
	Modification of **Figures 6-33** to **6-35**	
	Modification of **Figures 6-45, 6-49, 6-53, 6-57, 6-61, 6-65 Block Diagram**	
	Modification of **remark** in **6.8.3 Operation as multiple PWM output function**	
	Modification of **Figure 6-81. Procedure for Setting Remote control Output**	

(13/17)

Edition	Description	Chapter
Rev.1.00	Modification of **7.4.2 Shifting to HALT/STOP mode after starting operation**	**CHAPTER 7 REAL-TIME CLOCK**
	Modification of figure title in **Figure 7-23**	
	Modification of **Figure 8-5. 12-Bit Interval Timer Operation Timing (ITCMP11 to ITCMP0 = 0FFH, count clock: f_{SUB} = 32.768 kHz)**	**CHAPTER 8 INTERVAL TIMER**
	Addition of **9.3.3 Port mode registers 5, 14 (PM5, PM14)**	**CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER**
	Addition of **9.5 Cautions of clock output/buzzer output controller**	
	Modification of description in **10.1 Functions of Watchdog Timer**, **10.4.4 Setting watchdog timer interval interrupt**	**CHAPTER 10 WATCHDOG TIMER**
	Modification of **Figure 11-1. Block Diagram of A/D Converter**	**CHAPTER 11 A/D CONVERTER**
	Modification of error in **11.2 (9) AV_{REFP} pin**	
	Modification of **caution 1** in **11.3.1 Peripheral enable register 0 (PER0)**	
	Modification of **cautions 1** and **3** and addition of **caution 2** in **11.3.2 A/D converter mode register 0 (ADM0)**	
	Modification of **Table 11-1. Settings of ADCS and ADCE Bits**	
	Modification of description and addition of **note 2** and **caution 4** to **Figure 11-4. Timing Chart When A/D Voltage Comparator Is Used**	
	Modification of **Table 11-3. A/D Conversion Time Selection**	
	Modification of **cautions 1, 2** and addition of **caution 3** in **11.3.3 A/D converter mode register 1 (ADM1)**	
	Modification of description and **cautions 1** to **3** and addition of **note** in **Figure 11-7. Format of A/D Converter Mode register 2 (ADM2) (1/2)**	
	Modification of **caution** and addition of **note** and **remark** in **Figure 11-7. Format of A/D Converter Mode register 2 (ADM2) (2/2)**	
	Addition of **note** to **11.3.5 10-bit A/D conversion result register (ADCR)**, and **11.3.6 8-bit A/D conversion result register (ADCRH)**	
	Modification of **caution 5** and addition of **cautions 9, 10** in **11.3.7 Analog input channel specification register (ADS)**	
	Addition of **caution** to **11.3.10 A/D test register (ADTES)**	
	Addition of **caution 3** to **11.3.11 A/D port configuration register (ADPC)**	
	Addition of **caution** to **11.3.12 Port mode control registers 1, 4, 12, and 14 (PMC1, PMC4, PMC12, PMC14)**	
	Modification of description and addition of **caution** to **11.3.13 Port mode register 1, 2, 4, 12, and 14 (PM1, PM2, PM4, PM12, PM14)**	
	Addition of **note 1** to **11.4 A/D Converter Conversion Operations**	
	Modification of **Figures 11-26** to **11-28, 11-30**	
	Modification from **11.7.4 Setup when using temperature sensor (~)** to **11.7.4 Setup when temperature sensor output/internal reference voltage output is selected (~)** Modification from **Figure 11-29. Setup When Using Temperature Sensor** to **Figure 11-29. Setup when temperature sensor output/internal reference voltage output is selected**	

(14/17)

Edition	Description	Chapter
Rev.1.00	Modification of description in **11.8 SNOOZE Mode Function**	**CHAPTER 11 A/D CONVERTER**
	Addition of **caution** to **11.10 (2) Input range of ANI0, ANI1 and ANI16 to ANI23 pins**	
	Modification of value in **Table 11-6. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)**	
	Modification of description and addition of **note 1** to **12.1.1 3-wire serial I/O (CSI00, CSI01)**	**CHAPTER 12 SERIAL ARRAY UNIT**
	Modification of description in **Figure 12-6. Format of Serial Communication Operation Setting Register mn (SCRmn)**	
	Modification of **note** to **Figure 12-10. Format of Serial Channel Start Register m (SSm)**	
	Modification of **note** to **Figure 12-11. Format of Serial Channel Stop Register m (STm)**	
	Modification of **Figure 12-14. Format of Serial Output Register m (SOm)**	
	Modification of description in **Figure 12-16. Format of Serial Standby Control Register m (SSCm)**	
	Modification of description in **12.3.17 Port output mode register 1 (POM1)**	
	Addition of description to **12.3.18 Port mode register 1 (PM1)**	
	Modification of **note 1** in **12.5 Operation of 3-Wire Serial I/O (CSI00, CSI01) Communication**	
	Modification of description in **12.5.1 Master transmission**	
	Modification of description in **Figures 12-24 to 26, 32 to 34, 40 to 42, 48 to 50, 56 to 58, 62 to 64, 75 to77, 83 to 85, 87 (operation procedure)**	
	Modification of **Figures 12-27, 29, 37, 43, 45, 51, 53, 59, 65, 67 (timing chart)**	
	Modification of description in **12.5.2 Master reception**	
	Modification of description in **12.5.3 Master transmission/reception**	
	Modification of **note** to **12.5.4 Slave transmission**, **12.5.5 Slave reception**, **12.5.6 Slave transmission/reception**	
	Modification of description in **12.5.7 SNOOZE mode function**	
	Modification of **caution** in **Figures 12-69** and **12-71**	
	Modification of description in **12.6.1 UART transmission** and **12.6.2 UART reception**	
	Modification of **Figure 12-79, 81 (flow chart)**	
	Modification of **Figure 12-82 (Example of Contents of Registers)**.	
	Addition of description in **12.6.3 SNOOZE mode function**	
	Modification of **note** and **caution** in **Figure 12-88. Timing Chart of SNOOZE Mode Operation (Normal operation mode)**	
	Modification of **caution** in **Figure 12-89. Timing Chart of SNOOZE Mode Operation (Abnormal Operation <1>)**	
	Modification of **Figure 12-90. Flowchart of SNOOZE Mode Operation (Normal Operation/Abnormal Operation <1>)**	
	Modification of **note 1** and **caution 1** in **Figure 12-91. Timing Chart of SNOOZE Mode Operation (Abnormal Operation <2>)**	
	Modification of **Figure 12-92. Flowchart of SNOOZE Mode Operation (Abnormal Operation <2>)**	
	Modification of block diagram in **Figure 13-1. Block Diagram of LCD Controller/Driver**	**CHAPTER 13 LCD CONTROLLER/DRIVER**
	Deletion of note in **Figure 13-3. Format of LCD Mode Register 0 (LCDM0) (2/2)**	
	Modification of **Table 13-4. Combinations of Display Waveform, Time Slices, and Bias Method**	

(15/17)

Edition	Description	Chapter
Rev.1.00	Addition of **note 1** to **Figure 13-4. Format of LCD Mode Register 1 (LCDM1) (1/2)**	**CHAPTER 13 LCD CONTROLLER/DRIVER**
	Modification of **note** and **caution 1** in **Figure 13-4. Format of LCD Mode Register 1 (LCDM1) (2/2)**	
	Modification of **caution** in **Figure 13-5. Format of Operation Speed Mode Control Register (OSMC)**	
	Deletion of **note** and modification of **cautions 2, 3** in **Figure 13-6. Format of LCD Clock Control Register 0 (LCDC0)**	
	Modification of **caution** in **Figure 13-21. Operation Stop Procedure**	
	Modification of **caution** in **Figure 13-22. Examples of LCD Drive Power Connections (External Resistance Division Method)**	
	Modification of **Figure 13-23. Examples of LCD Drive Power Connections (Internal Voltage Boosting Method)**	
	Modification of **Figure 13-24. Examples of LCD Drive Power Connections (Capacitor Split Method)**	
	Modification of **Figure 13-29. Static LCD Drive Waveform Examples for SEG11, SEG12, and COM0**	
	Addition of description	**CHAPTER 14 INTERRUPT FUNCTION**
	Modification of **Table 14-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing**	
	Modification of all	**CHAPTER 15 KEY INTERRUPT FUNCTION**
	Modification **caution 1** in **16.1 Standby Function and Configuration**	**CHAPTER 16 STANDBY FUNCTION**
	Modification of **Table 16-1. Operating Statuses in HALT Mode**	
	Addition of **note 1** in **Figure 16-3. HALT Mode Release by Interrupt Request Generation**	
	Modification of description in **Figure 16-4. HALT Mode Release by Reset**	
	Modification of **caution 1** in **16.3.2 (1) STOP mode setting and operating statuses**	
	Modification of **Table 16-2. Operating Statuses in STOP Mode**	
	Modification of description in **Figure 16-5. STOP Mode Release by Interrupt Request Generation**	
	Modification of description in **Figure 16-6. STOP Mode Release by Reset**	
	Modification of description in **16.3.3 (1) SNOOZE mode setting and operating statuses**	
	Modification of description in **Table 16-3. Operating Statuses in SNOOZE Mode**	
	Modification of **caution 3** in **CHAPTER 17 RESET FUNCTION**	**CHAPTER 17 RESET FUNCTION**
	Modification of description in **Figure 17-2**	
	Modification and addition of **note 2** to **Figure 17-3**	
	Modification of port (latch) in **Table 17-1. Operation Statuses During Reset Period**	
	Modification of description of high-speed on-chip oscillator trimming register (HIOTRM) and **note 2** in **Table 17-2. Hardware Statuses After Reset Acknowledgment**	
	Modification of **caution 2** in **Figure 17-5. Format of Reset Control Flag Register (RESF)**	

(16/17)

Edition	Description	Chapter
Rev.1.00	Modification of description in **18.1 Functions of Power-on-reset Circuit**	**CHAPTER 18 POWER-ON-RESET CIRCUIT**
	Modification of description in **18.3 Operation of Power-on-reset Circuit**	
	Modification of description and **notes** in **Figure 18-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3) to (3/3)**	
	Modification of **Figure 19-1. Block Diagram of Voltage Detector**	**CHAPTER 19 VOLTAGE DETECTOR**
	Modification of **notes 1, 3** in **Figure 19-2. Format of Voltage Detection Register (LVIM)**	
	Modification of **Table 19-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H)**	
	Modification of **19.4.1 When used as reset mode**	
	Modification of **19.4.2 When used as interrupt mode**	
	Modification of description in **19.4.3 When used as interrupt and reset mode**	
	Modification of **Figure 19-6 (1/2)**	
	Modification of **Figure 19-6 (2/2)**	
	Modification of **Figure 20-3. Flowchart of Flash Memory CRC Operation Function (High-speed CRC)**	**CHAPTER 20 SAFETY FUNCTIONS**
	Addition of description and **caution** to **20.3.2 CRC operation function (general-purpose CRC)**	
	Modification of **caution** in **Figure 20-7. Format of RAM Parity Error Control Register (RPECTL)**	
	Modification of **Figure 20-10. Invalid Access Detection Area**	
	Addition of Figure (move from 2.2 Description to Pin Function (preceding editions))	**CHAPTER 21 REGULATOR**
	Modification of description and addition of **caution** to **22.1 Functions of Option Bytes**	**CHAPTER 22 OPTION BYTE**
	Modification of description in **Figure 22-1. Format of User Option Byte (000C0H)**	
	Modification of **Figure 22-2. Format of User Option Byte (000C1H)**	
	Deletion of description in **23.1.1 Programming environment**	**CHAPTER 23 FLASH MEMORY**
	Addition of description to **23.2 Writing to Flash Memory by Using External Device (that Incorporates UART)**	
	Addition of **remark** to **23.3 Connection of Pins on Board**	
	Addition of description, **caution** and **remark** to **23.4.1 Data flash overview**	
	Modification of **Figure 23-8. Setting of Flash Memory Programming Mode**	
	Modification of error in **Table 23-4. Relationship Between TOOL0 Pin and Operation Mode After Reset Release**	
	Modification of **Table 23-5. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified**	
	Modification of description in **Table 23-10. Example of Signature Data**	
	Addition of description and **caution** to **23.6 Security Settings**	
	Modification of **Table 23-11. Relationship Between Enabling Security Function and Command** and **Table 23-12. Setting Security in Each Programming Mode**	
	Modification of **cautions 2 to 4** to **23.7 Flash Memory Programming by Self-Programming**	
	Modification of **Table 23-13. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified**	
	Modification of **Table 23-14. Relationship Between Flash Shield Window Function Setting/Change Methods and Commands**	

R7F0C001G/L, R7F0C002G/L User’s Manual: Hardware

Publication Date: Rev.2.30 Jun 30, 2024

Published by: Renesas Electronics Corporation

R7F0C001G/L, R7F0C002G/L

Renesas Electronics Corporation