User's Manual

8

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

R7F0C80112ESP, R7F0C80212ESP

User's Manual: Hardware

8-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.1.00 Sep 2013

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended 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; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(2012.4)

NOTES FOR CMOS DEVICES

(1) 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 VIL (MAX) and VIH (MIN) due to noise, etc., 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 VIL (MAX) and VIH (MIN).

(2) HANDLING OF UNUSED INPUT PINS: Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

(3) PRECAUTION AGAINST ESD: A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that 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 should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

(4) STATUS BEFORE INITIALIZATION: Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

(5) POWER ON/OFF SEQUENCE: In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

(6) INPUT OF SIGNAL DURING POWER OFF STATE : Do not input signals or an I/O pull-up power supply while the device is not powered. 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. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the R7F0C80112ESP, R7F0C80212ESP and design and develop application systems and programs for these devices.

Purpose

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

Organization

The R7F0C80112ESP, R7F0C80212ESP manual is separated into two parts: this manual and the software edition (common to the RL78 family).

R7F0C80112ESP, R7F0C80212ESP **User's Manual** **Hardware**	**RL78 Family** **User's Manual** **Software**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

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

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler.
- To know details of the R7F0C80112ESP, R7F0C80212ESP Microcontroller instructions:
  → Refer to the separate document **RL78 Family Software User's Manual (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.
R7F0C80112ESP, R7F0C80212ESP User's Manual Hardware	This manual
RL78 Family Software User's Manual	R01US0015E

Documents Related to Flash Memory Programming

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

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

Other Documents

Document Name	Document No.
Renesas MPUs & MCUs RL78 Family	R01CP0003E
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E
Semiconductor Reliability Handbook	R51ZZ0001E

Note See the "Semiconductor Package Mount Manual" website (http://www.renesas.com/products/package/manual/index.jsp).

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 7 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER 163

CHAPTER 8 WATCHDOG TIMER 168

CHAPTER 9 A/D CONVERTER 173

CHAPTER 1 OUTLINE

1.1 Features

- Minimum instruction execution time can be changed from high speed (0.05 μs @ 20 MHz operation) to low speed (0.8 μs @ 1.25 MHz operation)
- General-purpose register: 8 bits × 8 registers
- ROM: 1 KB/2 KB, RAM: 128 bytes/256 bytes
- High-speed on-chip oscillator: 20/10/5/2.5/1.25 MHz (TYP) can be selected
- On-chip single-power-supply flash memory
- On-chip debug function
- On-chip selectable power-on-reset (SPOR) circuit
- On-chip watchdog timer (operable with the dedicated low-speed on-chip oscillator clock)
- On-chip key interrupt function: 6 key interrupt input pins
- On-chip clock output/buzzer output controller
- On-chip BCD adjustment
- I/O ports: 8
- Timer
  - 8-/16-bit timer: 2 channels
  - Watchdog timer: 1 channel
- Serial interface
  - CSI: 1 channel
  - UART: 1 channel
- 8/10-bit resolution A/D converter: 4 channels
- Standby function: HALT, or STOP mode
- <R> Power supply voltage: V_{DD} = 2.4 to 5.5 V
  (Use this product within the voltage range from 2.57 (TYP.) to 5.5 V because the detection voltage (V_{SPOR}) of the selectable power-on-reset (SPOR) circuit should also be considered.)
- Operating ambient temperature: T_A = –40 to +85°C
- ROM, RAM capacities

Flash ROM	RAM	Products Name
1 KB	128 B	R7F0C80112ESP
2 KB	256 B	R7F0C80212ESP

1.2 List of Part Numbers

Figure 1-1. Part Number, Memory Size, and Package of R7F0C80112ESP, R7F0C80212ESP

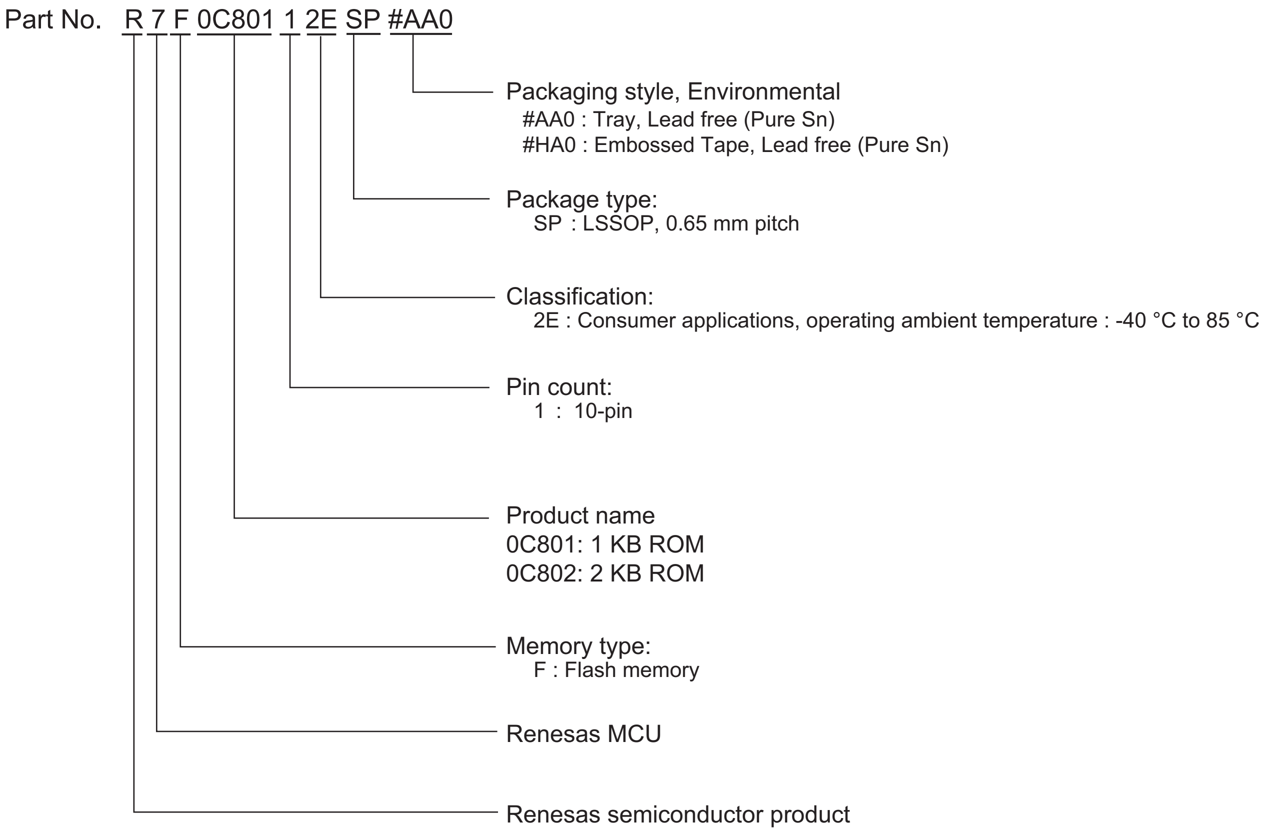
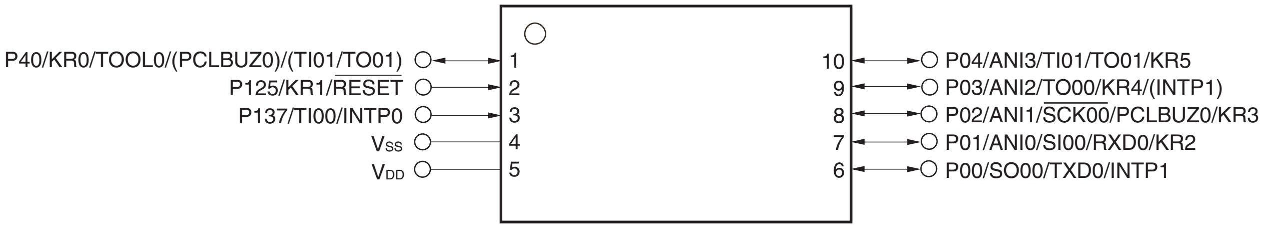


Table 1-1. List of Ordering Part Numbers

Pin count	Package	Flash ROM	RAM	Packaging style, Environmental	Part Number
10 pins	10-pin plastic LSSOP (4.4 × 3.6 mm, 0.65mm pitch)	1 KB	128 B	Tray Lead Free (Pure Sn)	R7F0C80112ESP#AA0
				Embossed Tape Lead Free (Pure Sn)	R7F0C80112ESP#HA0
		2 KB	256 B	Tray Lead Free (Pure Sn)	R7F0C80212ESP#AA0
				Embossed Tape Lead Free (Pure Sn)	R7F0C80212ESP#HA0

1.3 Pin Configuration (Top View)

- 10-pin plastic SSOP (4.4 × 3.6)



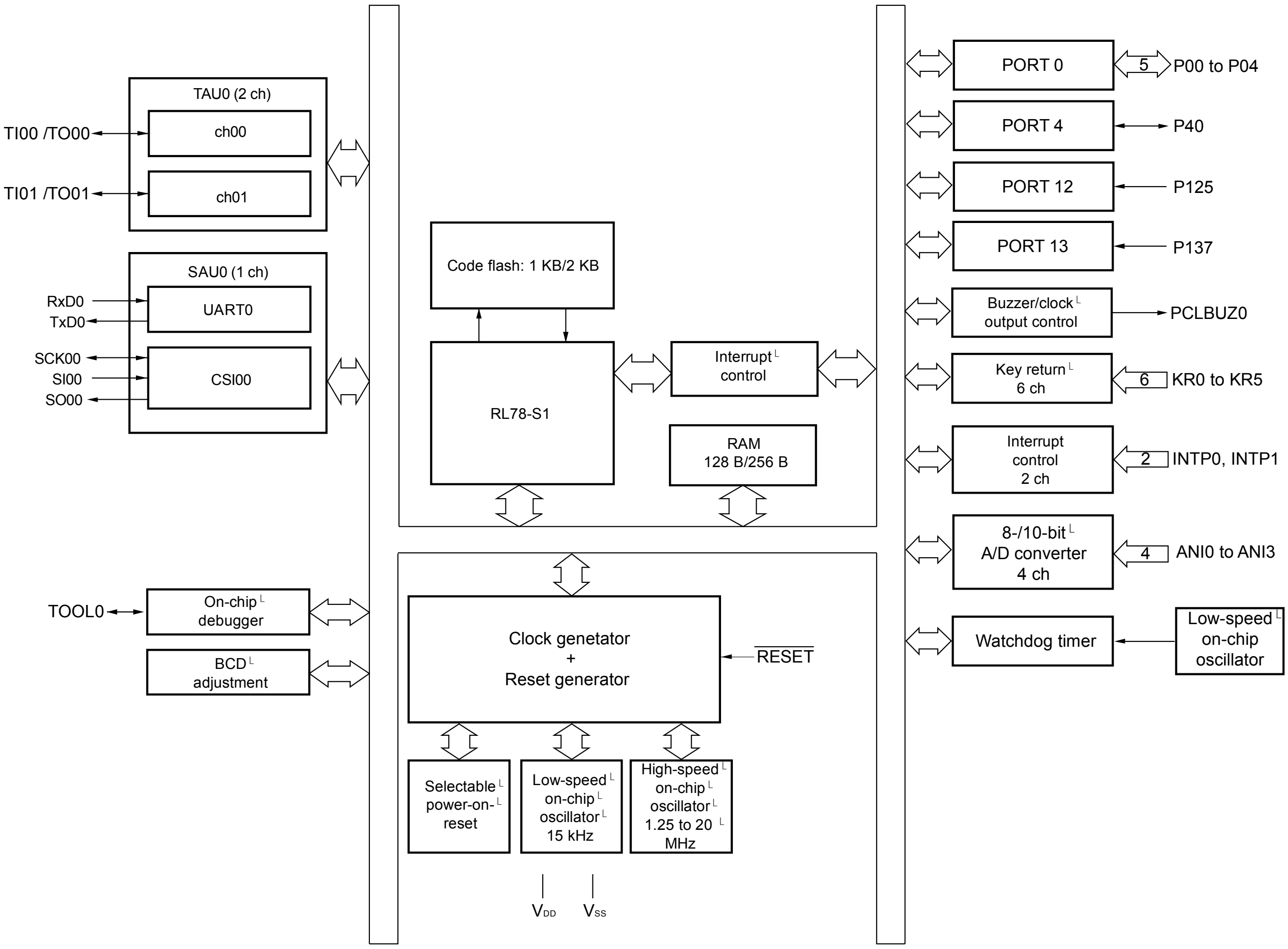
Remarks 1. For pin identification, see **1.4 Pin Identification**.

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

1.4 Pin Identification

ANI0 to ANI3 : Analog input
INTP0, INTP1 : External interrupt input
KR0 to KR5 : Key return
P00 to P04 : Port 0
P40 : Port 4
P125 : Port 12
P137 : Port 13
PCLBUZ0 : Programmable clock output/buzzer output
$\overline{\text{RESET}}$: Reset
RxD0 : Receive data
$\overline{\text{SCK00}}$: Serial clock input/output
SI00 : Serial data input
SO00 : Serial data output
TI00, TI01 : Timer input
TO00, TO01 : Timer output
TOOL0 : Data input/output for tool
TxD0 : Transmit data
V_{DD} : Power supply
V_{SS} : Ground

1.5 Block Diagram



1.6 Outline of Functions

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

Item		R7F0C80212ESP	R7F0C80112ESP
Code flash memory		2 KB	1 KB
RAM		256 B	128 B
Main system clock	High-speed on-chip oscillator clock	• 1.25 to 20 MHz (V_{DD} = 2.7 to 5.5 V) • 1.25 to 5 MHz (V_{DD} = 2.4 to 5.5 V)	
Low-speed on-chip oscillator clock		15 kHz (TYP)	
General-purpose register		8 bits register × 8	
Minimum instruction execution time		0.05 μs (20 MHz operation)	
Instruction set		• Data transfer (8 bits) • Adder and subtractor/logical operation (8 bits) • Multiplication (8 bits × 8 bits) • Rotate, barrel shift, and bit manipulation (set, reset, test, and Boolean operation), etc.	
I/O port	Total	8	
	CMOS I/O	6 (N-ch open-drain output (V_{DD} tolerance): 1)	
	CMOS input	2	
Timer	16-bit timer	2 channels	
	Watchdog timer	1 channel	
	Timer output	2 channels (PWM output: 1)	
Clock output/buzzer output		1	
		2.44 kHz to 10 MHz: (Peripheral hardware clock: f_{MAIN} = 20 MHz operation)	
8-/10-bit resolution A/D converter		4 channels	
Serial interface		CSI: 1 channel/UART: 1 channel	
Vectored interrupt sources	Internal	8	
	External	3	
Key interrupt		6	
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by selectable power-on-reset <R> • Internal reset by illegal instruction execution[Note 1]	
Selectable power-on-reset circuit		• Detection voltage: Rising edge (V_{SPOR}): 2.27 V/2.90 V/4.28 V (max.) Falling edge (V_{SPDR}): 2.40 V/2.70 V/4.00 V (max.)	
On-chip debug function		Provided	
Power supply voltage		V_{DD} = 2.4 to 5.5 V[Note 2]	
Operating ambient temperature		T_A = –40 to + 85 °C	

Note 1. This reset occurs when instruction code FFH is executed. This reset does not occur during emulation using an in-circuit emulator or an on-chip debugging emulator.

<R> **2.** Use this product within the voltage range from 2.57 to 5.5 V because the detection voltage (V_{SPOR}) of the selectable power-on-reset (SPOR) circuit should also be considered.

CHAPTER 2 PIN FUNCTIONS

2.1 Port Functions

The input or output, buffer, and pull-up resistor settings are also valid for the alternate functions.

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 5-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input. When the each pin is used as input, specify them as either digital or analog in port mode control register 0 (PMC0). This register can be specified in 1-bit units. Output from P00 can be specified as N-ch open-drain (V_{DD} tolerant). At input port use of an on-chip pull-up resistor can be specified by a software setting (1-bit units).	Input port	SO00/TXD0/INTP1
P01			Analog input port	ANI0/SI00/RXD0/KR2
P02				ANI1/$\overline{\text{SCK00}}$/PCLBUZ0/KR3
P03				ANI2/TO00/KR4/(INTP1)
P04				ANI3/TI01/TO01/KR5
P40	I/O	Port 4. 1-bit I/O port. Input/output can be specified. At input port use of an on-chip pull-up resistor can be specified by a software setting.	Input port	KR0/TOOL0/(PCLBUZ0)/(TI01/TO01)
P125	Input	Port 12 1-bit input port. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	KR1/$\overline{\text{RESET}}$
P137	Input	Port 13 1-bit input port	Input port	TI00/INTP0

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Register (PIOR)**.

2.2 Functions other than port pins

<table>
<tr><th>Function Name</th><th>I/O</th><th>Functions</th></tr>
<tr><td>ANI0 to ANI3</td><td>Input</td><td>Analog input pins of A/D converter
(See, Figure 9-22. Analog Input Pin Connection.)</td></tr>
<tr><td>INTP0, INTP1</td><td>Input</td><td>External interrupt request input
Specified available edge: rising edge, falling edge, or both rising and falling edges</td></tr>
<tr><td>KR0 to KR5</td><td>Input</td><td>Key interrupt input</td></tr>
<tr><td>PCLBUZ0</td><td>Output</td><td>Clock/buzzer output</td></tr>
<tr><td>$\overline{\text{RESET}}$</td><td>Input</td><td>External reset input
When the external reset pin is not used, leave open, or connect to V_{DD} while PORTSELB = 1.</td></tr>
<tr><td>RxD0</td><td>Input</td><td>UART0 serial data input</td></tr>
<tr><td>TxD0</td><td>Output</td><td>UART0 serial data output</td></tr>
<tr><td>$\overline{\text{SCK00}}$</td><td>I/O</td><td>CSI00 clock I/O</td></tr>
<tr><td>SI00</td><td>Input</td><td>CSI00 serial data input</td></tr>
<tr><td>SO00</td><td>Output</td><td>CSI00 serial data output</td></tr>
<tr><td>TI00, TI01</td><td>Input</td><td>Inputting an external count clock/capture trigger to 16-bit timers 00, 01</td></tr>
<tr><td>TO00, TO01</td><td>Output</td><td>Timer output pins of 16-bit timers 00, 01</td></tr>
<tr><td>V_{DD}</td><td>–</td><td>Positive power supply</td></tr>
<tr><td><R> V_{SS}</td><td>–</td><td>Ground potential</td></tr>
<tr><td>TOOL0</td><td>I/O</td><td>Data I/O pin for a flash memory programmer/debugger
The operation mode after start-up is determined by the status of the TOOL0 pin at the time of releasing a reset. Connect this pin via an external resistor to V_{DD} when enabling on-chip debugging (pulling it down is prohibited).
<table>
<tr><th>TOOL0</th><th>Operation mode</th></tr>
<tr><td>V_{DD}</td><td>Normal operation mode</td></tr>
<tr><td>0 V</td><td>Flash memory programming mode</td></tr>
</table>
For details, see 17.4.2 Flash memory programming mode.</td></tr>
</table>

2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

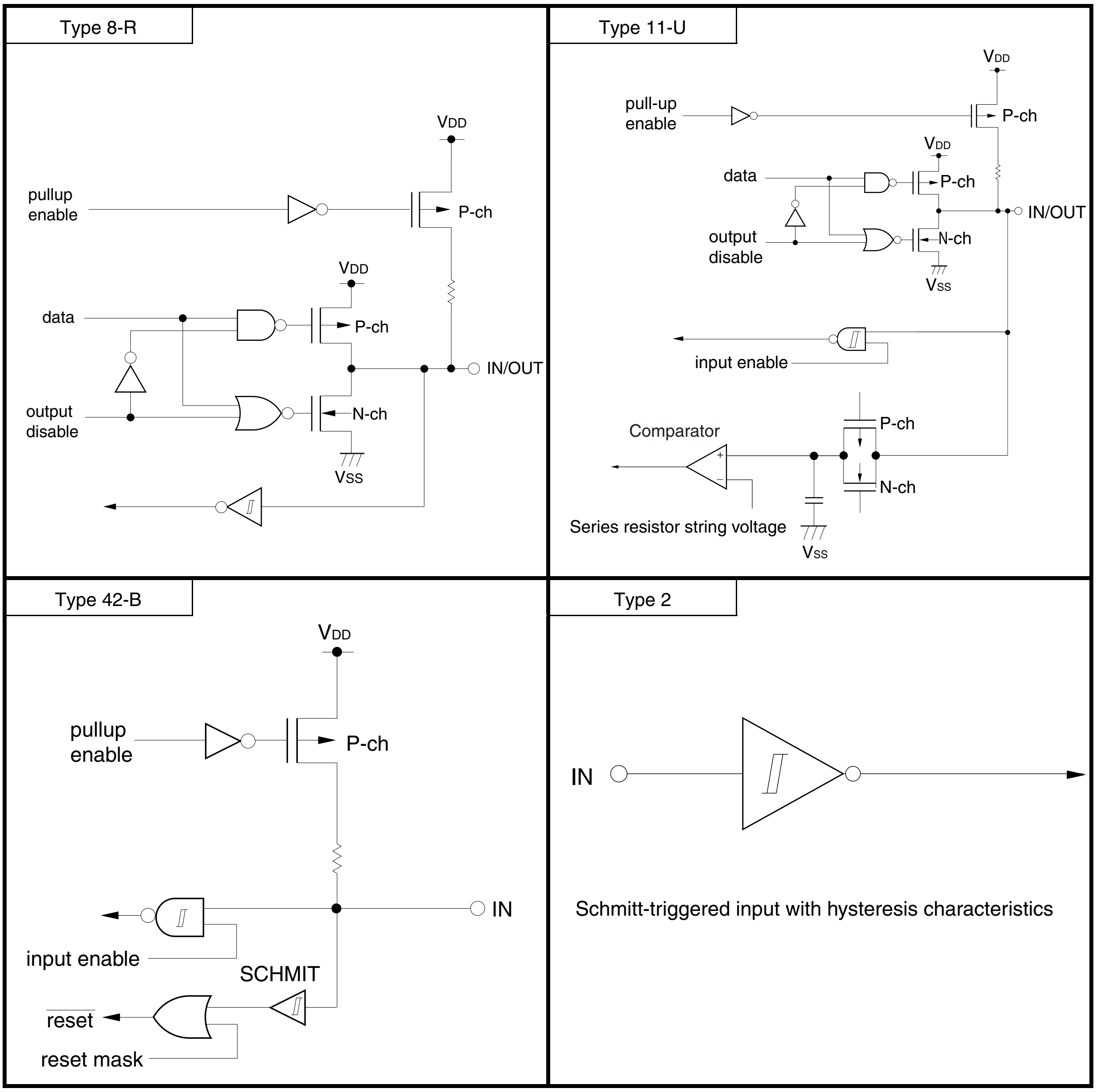
Tables 2-1 and 2-2 show the types of pin I/O circuits and the recommended connections of unused pins.

Table 2-1. Connection of Unused Pins

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/SO00/TXD0/INTP1	8-R	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P01/ANI0/SI00/RXD0/KR2	11-U		
P02/ANI1/$\overline{\text{SCK00}}$/PCLBUZ0/KR3			
P03/ANI2/TO00/KR4/(INTP1)			
P04/ANI3/TI01/TO01/KR5			
P40/KR0/TOOL0/(PCLBUZ0)/ (TI01/TO01)	8-R		Independently connect to V_{DD} via a resistor.
P125/KR1/$\overline{\text{RESET}}$	42-B	input	Leave open, or connect to V_{DD} while PORTSELB = 1.
P137/INTP0	2		Independently connect to V_{DD} or V_{SS} via a resistor.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Register (PIOR)**.

Figure 2-1. Pin I/O Circuit List



CHAPTER 3 CPU ARCHITECTURE

<R> R7F0C80112ESP, R7F0C80212ESP have the RL78-S1 core.

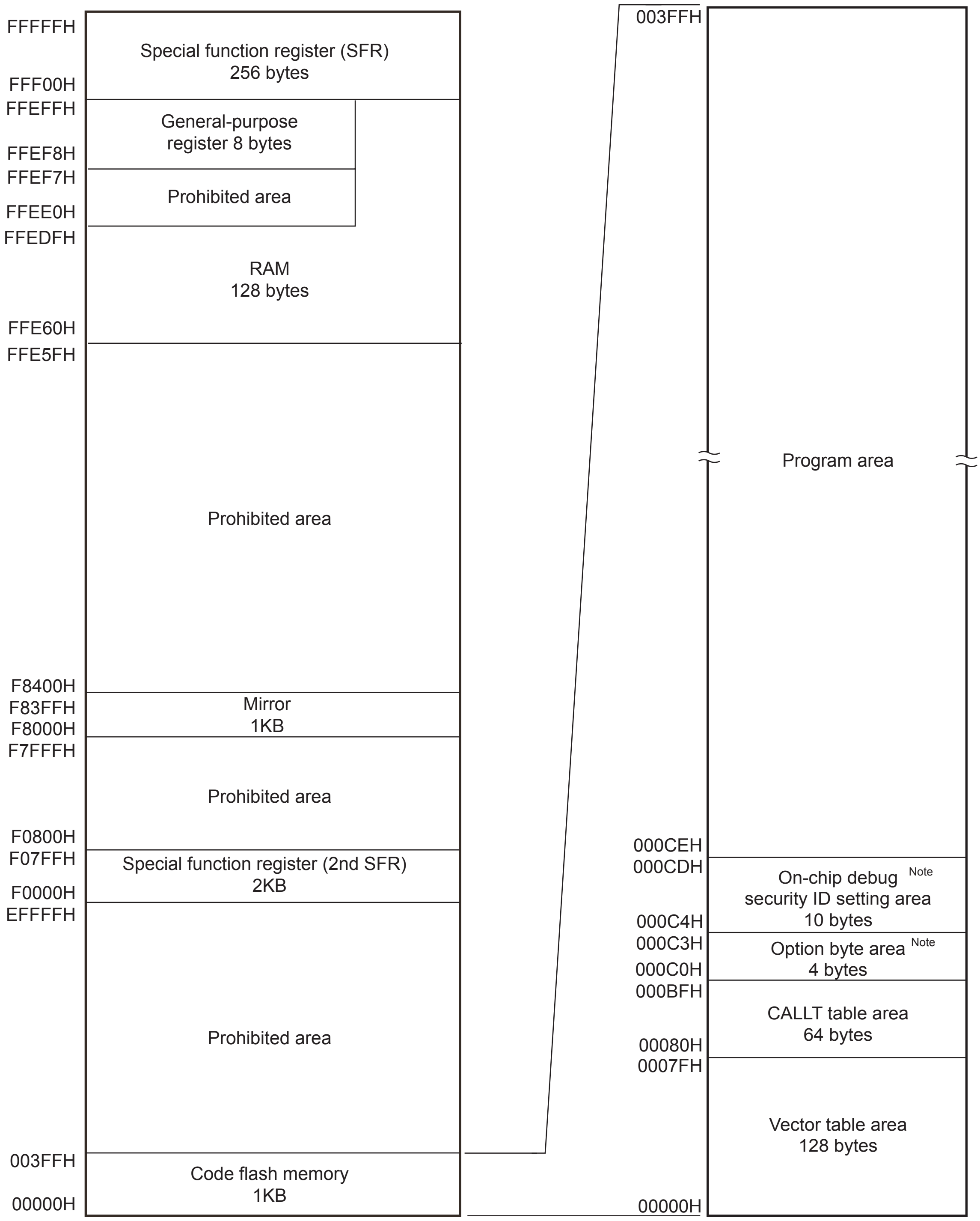
The features of the RL78-S1 core are as follows.

- CISC architecture with 3-stage pipeline
- Address space: 1 MB
- General-purpose register: 8-bit register x 8
- The RL78-S2 and RL78-S1 cores have a common instruction set. Note, however, the following instructions require a different number of clock cycles. For details, see **CHAPTER 20 INSTRUCTION SET**.
    - 16-bit data transfer (MOVW, XCHW, ONEW, CLRW)
    - 16-bit operation (ADDW, SUBW, CMPW)
    - Multiply (MULU)
    - 16-bit increment/decrement (INCW, DECW)
    - 16-bit shift (SHRW, SHLW, SARW)
    - 16-bit rotate (ROLWC)
    - Call/return (CALL, CALLT, BRK, RET, RETI, RETB)
    - Stack manipulate (PUSH, POP, MOVW, ADDW, SUBW)

3.1 Address Space

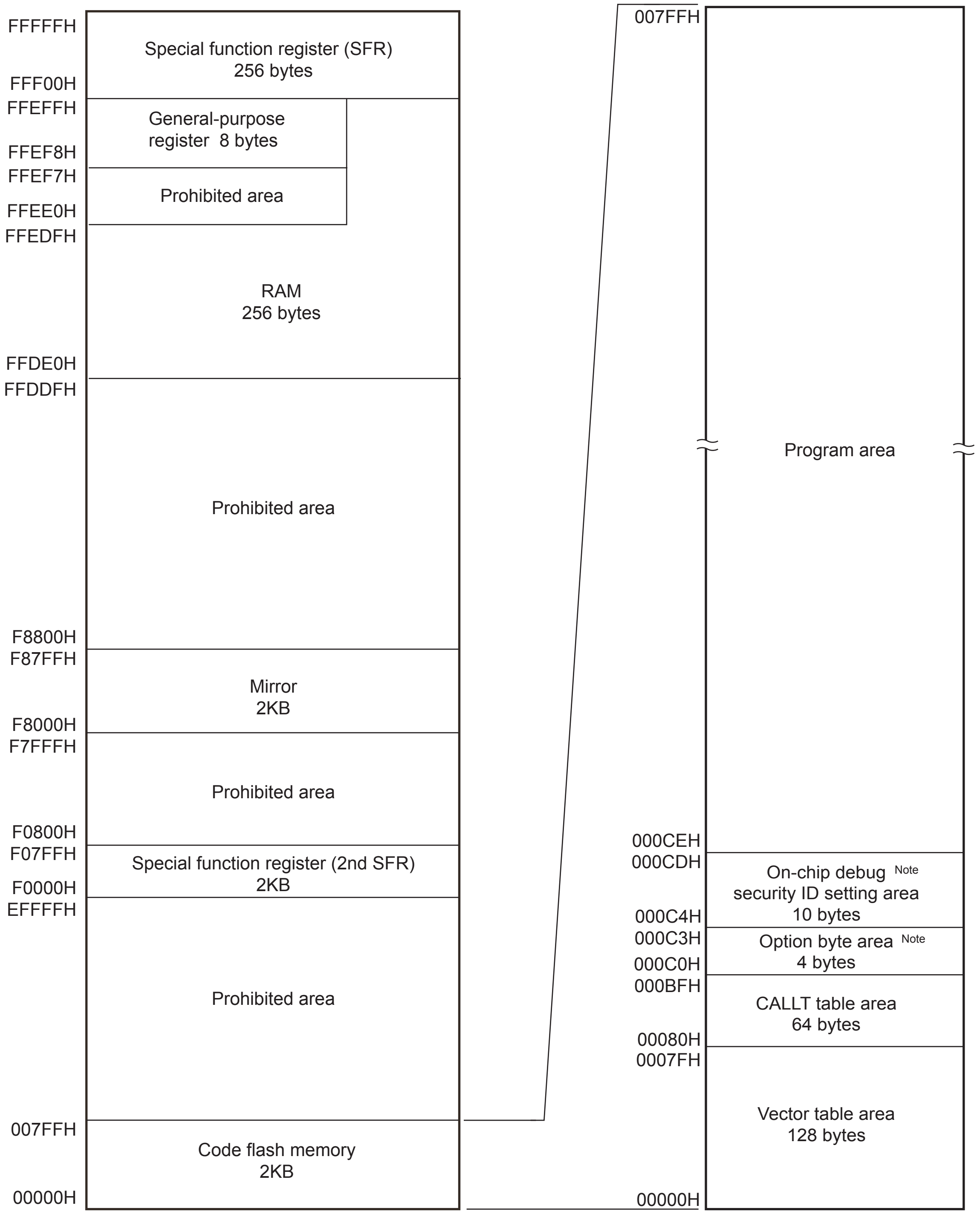
Products in the R7F0C80112ESP, R7F0C80212ESP can access a 1 MB address space. Figures 3-1 and 3-2 show the memory maps.

Figure 3-1. Memory Map for the R7F0C80112ESP



<R> **Caution Access to the reserved area is prohibited.**

Figure 3-2. Memory Map for the R7F0C80212ESP



<R> **Caution Access to the reserved area is prohibited.**

3.1.1 Internal program memory space

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

The R7F0C80112ESP, R7F0C80212ESP products incorporate internal ROM (flash memory), as shown below.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R7F0C80112ESP	Flash memory	1024 × 8 bits (00000H to 003FFH)
R7F0C80212ESP		2048 × 8 bits (00000H to 007FFH)

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

(1) Vector table area

The 128-byte area of 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 addresses are assigned to a 64 KB address area of 00000H to 0FFFFH, because the vector code is 2 bytes.

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

Table 3-2. Vector Table

Vector Table Address	Interrupt Source
00002H	–
00004H	INTWDTI
00006H	INTP0
00008H	INTP1
0000AH	INTST0, INTCSI00
0000CH	INTSR0
0000EH	INTSRE0
00010H	INTTM01H
00012H	INTTM00
00014H	INTTM01
00016H	INTAD
00018H	INTKR
0007EH	BRK

(2) CALLT instruction table area

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

(3) Option byte area

The 4-byte area of 000C0H to 000C3H can be used as an option byte area. For details, see **CHAPTER 16 OPTION BYTE**.

(4) On-chip debug security ID setting area

The 10-byte areas of 000C4H to 000CDH and 010C4H to 010CDH can be used as an on-chip debug security ID setting area. For details, see **CHAPTER 18 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The products with 1 KB /2 KB flash memory mirror the code flash area of 00000H to 003FFH/007FFH to the area of F8000H to F83FFH/F87FFH (the code flash area to be mirrored is set by the processor mode control register (PMC)).

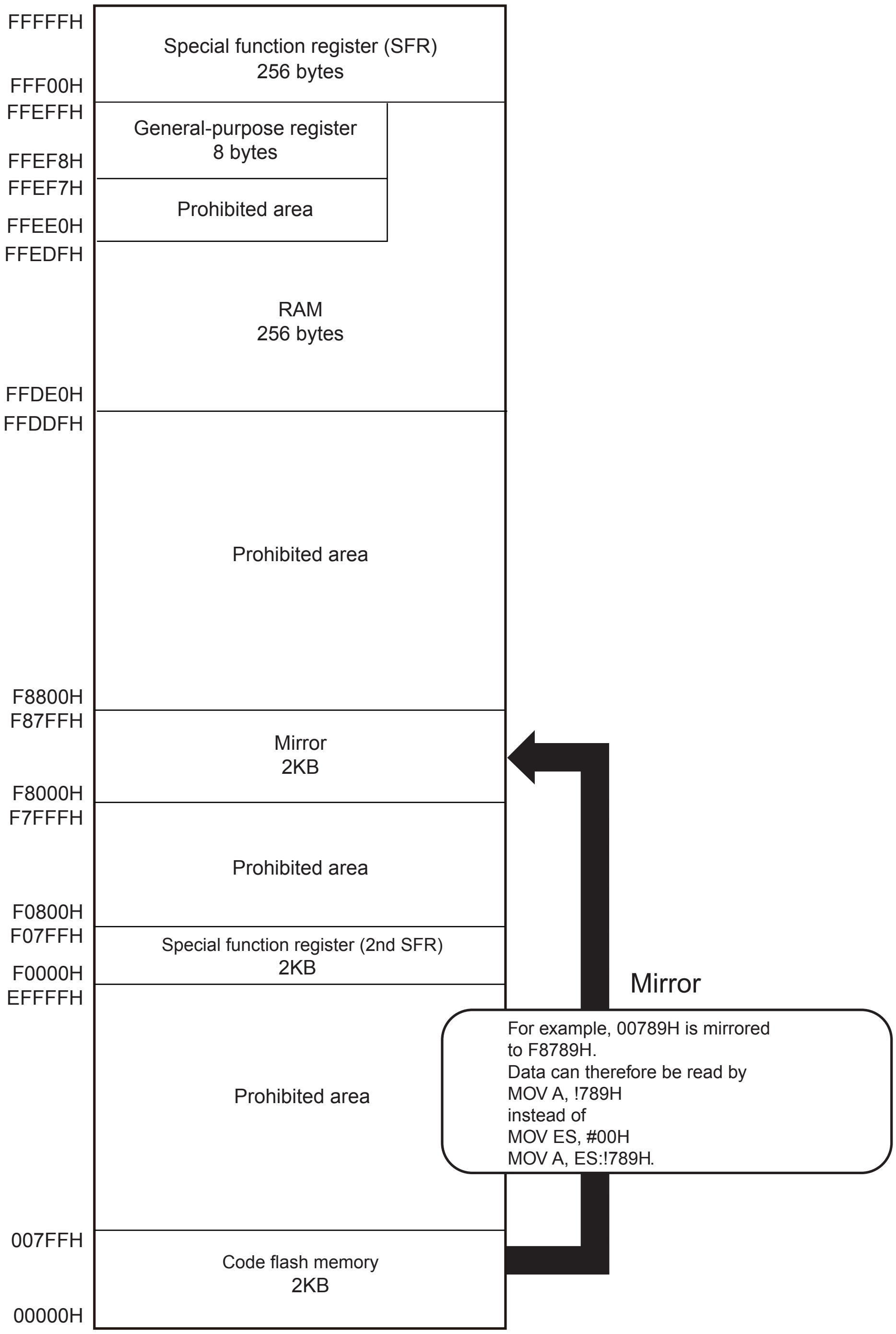
By reading data from F8000H to F83FFH/F87FFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code.

See **3.1 Address 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 shows examples.

Example R7F0C80112ESP (Flash memory: 2 KB)



For example, 00789H is mirrored to F8789H.
Data can therefore be read by
MOV A, !789H
instead of
MOV ES, #00H
MOV A, ES:!789H.

3.1.3 Internal data memory space

The R7F0C80112ESP, R7F0C80212ESP products incorporate the following RAMs.

Table 3-3. Internal RAM Capacity

Part Number	Internal RAM
R7F0C80112ESP	128 bytes (FFE60H to FFEDFH)
R7F0C80212ESP	256 bytes (FFDE0H to FFEDFH)

The internal RAM can be used as a data area and a program area where instructions are written and executed.

The internal RAM is used as a stack memory.

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area of FFF00H to FFFFFH (see **Table 3-4** 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 of F0000H to F07FFH (see **Table 3-5** 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

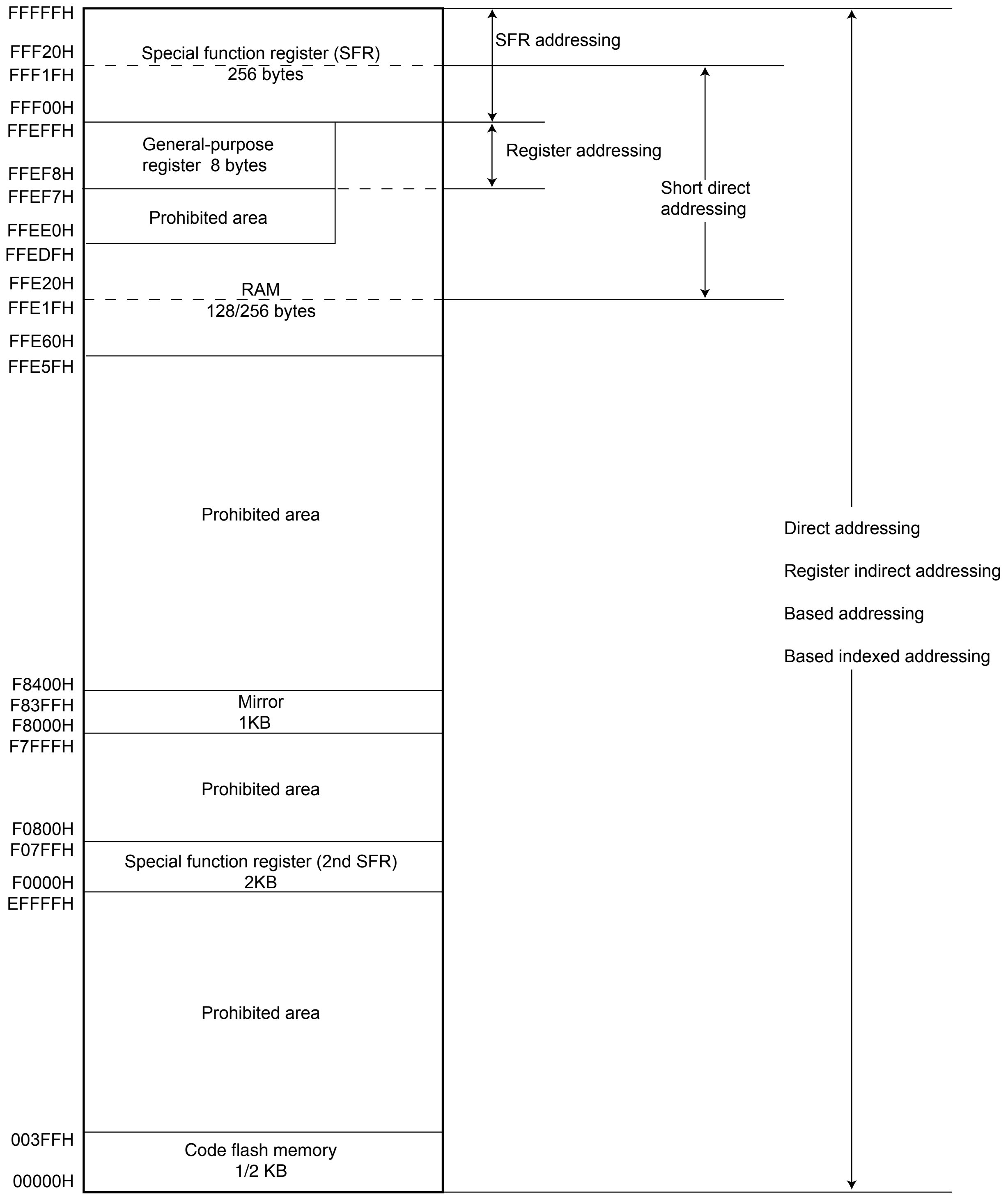
Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the R7F0C80112ESP, R7F0C80212ESP, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figures 3-3 show correspondence between data memory and addressing.

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

<R>

Figure 3-3. Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The R7F0C80112ESP, R7F0C80212ESP products incorporate the following processor registers.

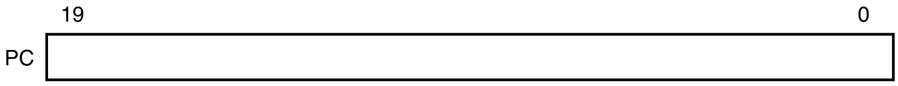
3.2.1 Control registers

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

(1) Program counter (PC)

The program counter is a 20-bit register that holds the address information of the next program to be executed.
In normal operation, PC is automatically incremented according to the number of bytes of the instruction to be fetched.
When a branch instruction is executed, immediate data and register contents are set.
Reset signal generation sets the reset vector table values at addresses 0000H and 0001H to the 16 lower-order bits of the program counter. The four higher-order bits of the program counter are cleared to 0000.

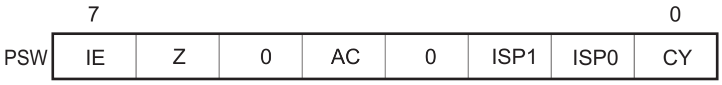
Figure 3-4. Format of Program Counter



(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 acknowledgment of a vectored interrupt request 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 3-5. Format of Program Status Word



	7							0
PSW	IE	Z	0	AC	0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

This flag controls the interrupt request acknowledge operations of the CPU.
When 0, the IE flag is set to the interrupt disabled (DI) state, and all maskable interrupt requests are disabled.
When 1, the IE flag is set to the interrupt enabled (EI) state, and interrupt request acknowledgment is controlled with an in-service priority flag (ISP1, ISP0), an interrupt mask flag for various interrupt sources, and a priority specification flag.
The IE flag is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (1) upon EI instruction execution.

(b) Zero flag (Z)

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

(c) 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.

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

These flags manage the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PR00L, PR00H, PR10L, PR10H) (see **11.3.3 Priority specification flag registers (PR00L, PR00H, PR10L, PR10H)**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

(e) 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 3-6. Format of Stack Pointer

<R>

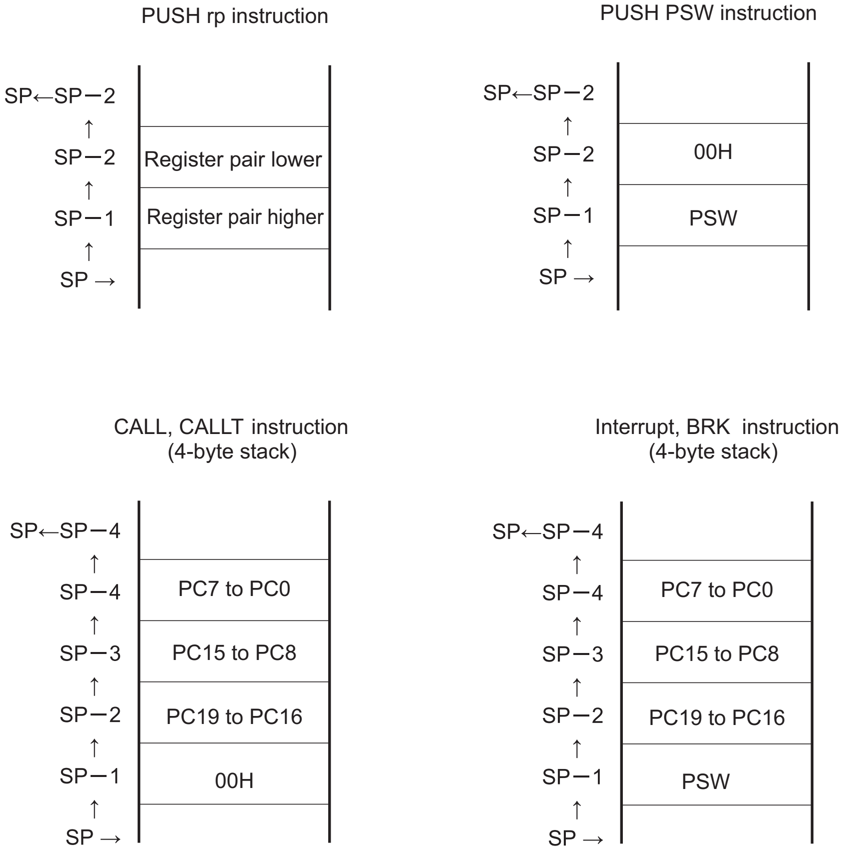
	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	0

The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

Each stack operation saves data as shown in Figure 3-7.

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

Figure 3-7. Data to Be Saved to Stack Memory



3.2.2 General-purpose registers

The general-purpose registers are a bank of eight 8-bit registers (X, A, C, B, E, D, L, and H) mapped to addresses (FFEF8H to FFEFFH) of the data memory.

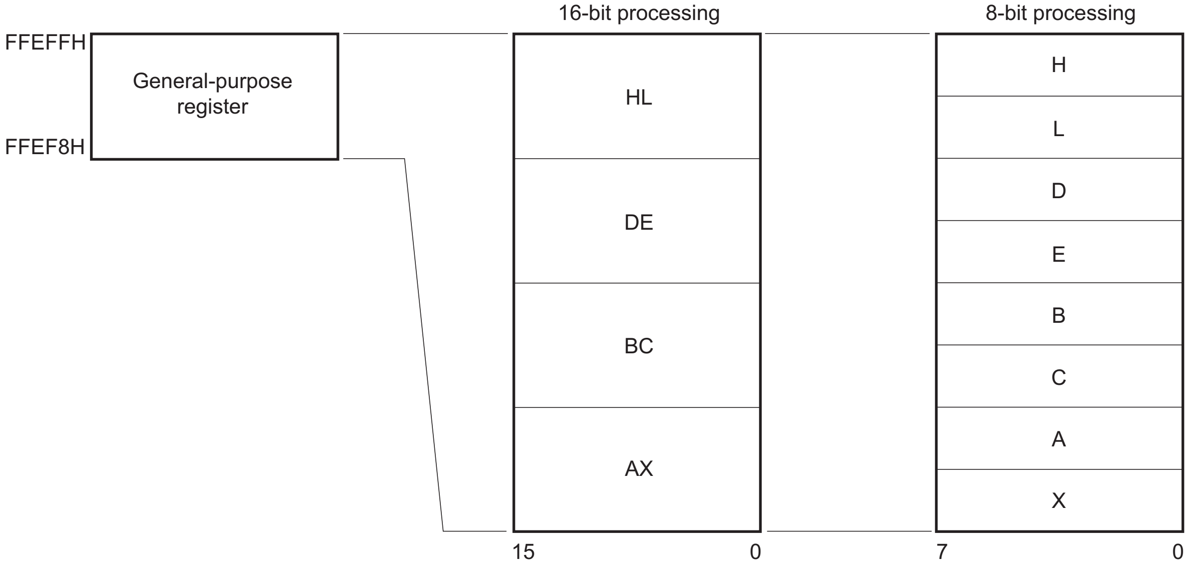
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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

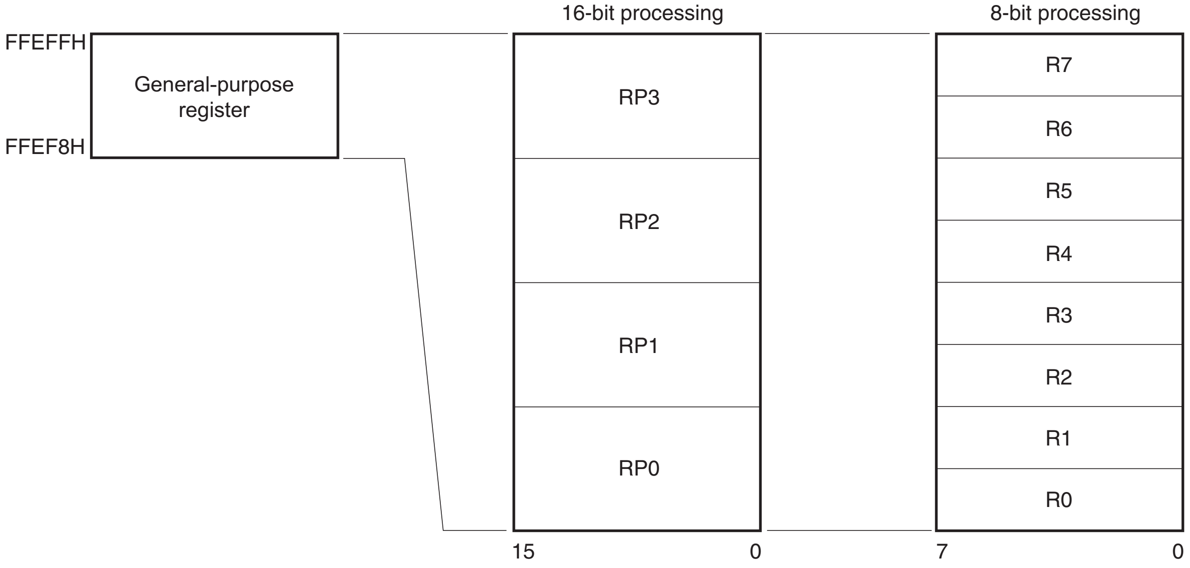
Caution It is prohibited to use the general-purpose register (FFEF8H to FFEFFH) space for fetching instructions or as a stack area.

Figure 3-8. Configuration of General-Purpose Registers

(a) Function name



(b) Absolute name



<R> 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

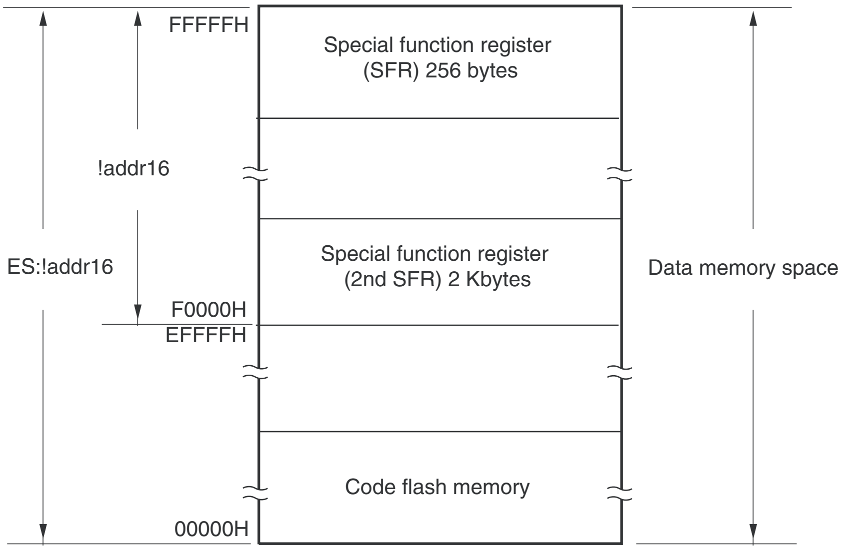
	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CP2	CP1	CP0

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

!addr16 → F 0000H - FFFFH

ES:!addr16 → 0H - FH 0000H - FFFFH



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 and 8, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit). This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (sfr). This manipulation can also be specified with an address.

Table 3-4 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 or 8). "–" 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 3-4. SFR List (1/2)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range		After Reset
					1-bit	8-bit	
FFF00H	Port register 0	P0		R/W	√	√	00H
FFF04H	Port register 4	P4		R/W	√	√	00H
FFF0CH	Port register 12	P12		R	√	√	Undefined
FFF0DH	Port register 13	P13		R	√	√	Undefined
FFF10H	Serial data register 00L	TXD0/ SIO00	SDR00L	R/W	–	√	00H
FFF11H	Serial data register 00H	–	SDR00H	R/W	–	√	00H
FFF12H	Serial data register 01L	RXD0/ SIO01	SDR01L	R/W	–	√	00H
FFF13H	Serial data register 01H	–	SDR01H	R/W	–	√	00H
FFF18H	Timer data register 00L	TDR00L		R/W	–	√	00H
FFF19H	Timer data register 00H	TDR00H		R/W	–	√	00H
FFF1AH	Timer data register 01L	TDR01L		R/W	–	√	00H
FFF1BH	Timer data register 01H	TDR01H		R/W	–	√	00H
FFF1EH	A/D conversion result lower bit register	ADCRL		R	–	√	00H
FFF1FH	A/D conversion result upper bit register	ADCRH		R	–	√	00H
FFF20H	Port mode register 0	PM0		R/W	√	√	FFH
FFF24H	Port mode register 4	PM4		R/W	√	√	FFH
FFF30H	A/D converter mode register 0	ADM0		R/W	√	√	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	00H
FFF34H	Key interrupt control register	KRCTL		R/W	√	√	00H
FFF35H	Key interrupt flag register	KRF		R/W	–	√	00H
FFF37H	Key interrupt mode register 0	KRM0		R/W	√	√	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	00H

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range		After Reset
				1-bit	8-bit	
FFFA5H	Clock output select register 0	CKS0	R/W	√	√	00H
FFFA8H	Reset control flag register	RESF	R	–	√	Undefined[Note 1]
FFFABH	Watchdog timer enable register	WDTE	R/W	–	√	1AH/9AH[Note 2]
FFFE0H	Interrupt request flag register 0L	IF0L	R/W	√	√	00H
FFFE1H	Interrupt request flag register 0H	IF0H	R/W	√	√	00H
FFFE4H	Interrupt mask flag register 0L	MK0L	R/W	√	√	FFH
FFFE5H	Interrupt mask flag register 0H	MK0H	R/W	√	√	FFH
FFFE8H	Priority specification flag register 00L	PR00L	R/W	√	√	FFH
FFFE9H	Priority specification flag register 00H	PR00H	R/W	√	√	FFH
FFFECH	Priority specification flag register 10L	PR10L	R/W	√	√	FFH
FFFEDH	Priority specification flag register 10H	PR10H	R/W	√	√	FFH

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

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

Remark For extended SFRs (2nd SFRs), see **Table 3-5 Extended SFR (2nd SFR) List**.

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

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

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

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

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (!addr16.bit). This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (!addr16). This manipulation can also be specified with an address.

Table 3-5 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 or 8). "–" 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 SFRs in the SFR area, see **3.2.4 Special function registers (SFRs)**.

Table 3-5. Extended SFR (2nd SFR) List (1/2)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range		After Reset
				1-bit	8-bit	
F0010H	A/D converter mode register 2	ADM2	R/W	√	√	00H
F0030H	Pull-up resistor option register 0	PU0	R/W	√	√	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	01H
F003CH	Pull-up resistor option register 12	PU12	R/W	√	√	00H
F0050H	Port output mode register 0	POM0	R/W	√	√	00H
F0060H	Port mode control register 0	PMC0	R/W	√	√	FFH
F0070H	Noise filter enable register 0	NFEN0	R/W	√	√	00H
F0071H	Noise filter enable register 1	NFEN1	R/W	√	√	00H
F0073H	Input switch control register	ISC	R/W	√	√	00H
F0077H	Peripheral I/O redirection register	PIOR	R/W	–	√	00H
F00A8H	High-speed on-chip oscillator trimming register	HOCODIV	R/W	–	√	Undefined
F00F0H	Peripheral enable register 0	PER0	R/W	√	√	00H
F00FEH	BCD adjust result register	BCDADJ	R	–	√	Undefined
F0100H	Serial status register 00	SSR00	R	–	√	00H
F0102H	Serial status register 01	SSR01	R	–	√	00H
F0108H	Serial flag clear trigger register 00	SIR00	R/W	–	√	00H
F010AH	Serial flag clear trigger register 01	SIR01	R/W	–	√	00H
F0110H	Serial mode register 00L	SMR00L	R/W	–	√	20H
F0111H	Serial mode register 00H	SMR00H	R/W	–	√	00H
F0112H	Serial mode register 01L	SMR01L	R/W	–	√	20H
F0113H	Serial mode register 01H	SMR01H	R/W	–	√	00H
F0118H	Serial communication operation setting register 00L	SCR00L	R/W	–	√	87H
F0119H	Serial communication operation setting register 00H	SCR00H	R/W	–	√	00H
F011AH	Serial communication operation setting register 01L	SCR01L	R/W	–	√	87H
F011BH	Serial communication operation setting register 01H	SCR01H	R/W	–	√	00H
F0120H	Serial channel enable status register 0	SE0	R	√	√	00H
F0122H	Serial channel start register 0	SS0	R/W	√	√	00H
F0124H	Serial channel stop register 0	ST0	R/W	√	√	00H
F0126H	Serial clock select register 0	SPS0	R/W	–	√	00H
F0128H	Serial output register 0	SO0	R/W	–	√	03H
F0129H	Serial clock output register 0	CKO0	R/W	–	√	03H
F012AH	Serial output enable register 0	SOE0	R/W	√	√	00H
F0134H	Serial output level register 0	SOL0	R/W	–	√	00H
F0180H	Timer counter register 00L	TCR00L	R	–	√	FFH
F0181H	Timer counter register 00H	TCR00H	R	–	√	FFH
F0182H	Timer counter register 01L	TCR01L	R	–	√	FFH
F0183H	Timer counter register 01H	TCR01H	R	–	√	FFH

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range		After Reset
				1-bit	8-bit	
F0190H	Timer mode register 00L	TMR00L	R/W	–	√	00H
F0191H	Timer mode register 00H	TMR00H	R/W	–	√	00H
F0192H	Timer mode register 01L	TMR01L	R/W	–	√	00H
F0193H	Timer mode register 01H	TMR01H	R/W	–	√	00H
F01A0H	Timer status register 00	TSR00	R	–	√	00H
F01A2H	Timer status register 01	TSR01	R	–	√	00H
F01B0H	Timer channel enable status register 0	TE0	R	√	√	00H
F01B1H	Timer channel enable status register 0 (8-bit mode)	TEH0	R	√	√	00H
F01B2H	Timer channel start register 0	TS0	R/W	√	√	00H
F01B3H	Timer channel start register 0 (8-bit mode)	TSH0	R/W	√	√	00H
F01B4H	Timer channel stop register 0	TT0	R/W	√	√	00H
F01B5H	Timer channel stop register 0 (8-bit mode)	TTH0	R/W	√	√	00H
F01B6H	Timer clock select register 0	TPS0	R/W	–	√	00H
F01B8H	Timer output register 0	TO0	R/W	–	√	00H
F01BAH	Timer output enable register 0	TOE0	R/W	√	√	00H
F01BCH	Timer output level register 0	TOL0	R/W	–	√	00H
F01BEH	Timer output mode register 0	TOM0	R/W	–	√	00H

Remark For SFRs in the SFR area, see **Table 3-4 SFR List**.

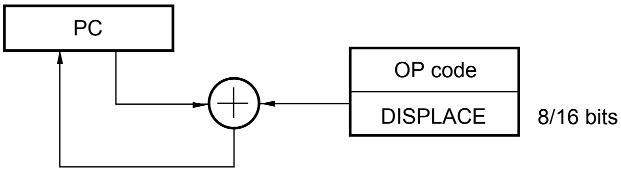
3.3 Instruction Address Addressing

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

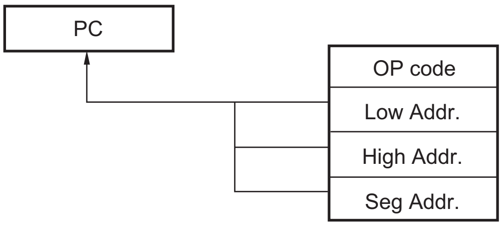
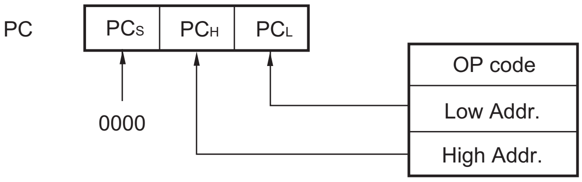


Figure 3-13. Example of CALL !addr16/BR !addr16



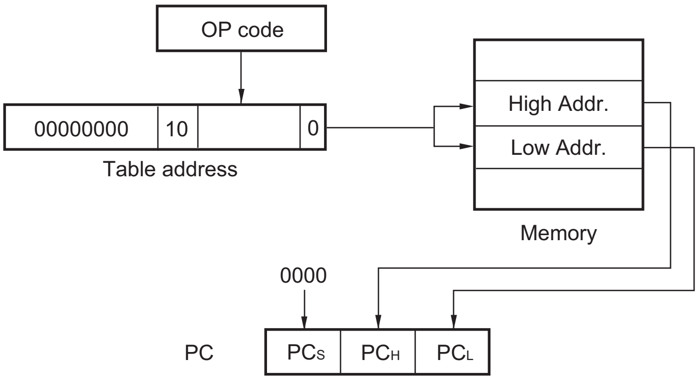
3.3.3 Table indirect addressing

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

Figure 3-14. Outline of Table Indirect Addressing

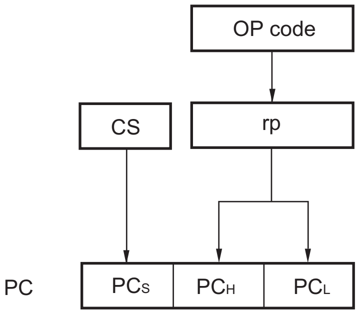


3.3.4 Register direct addressing

[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

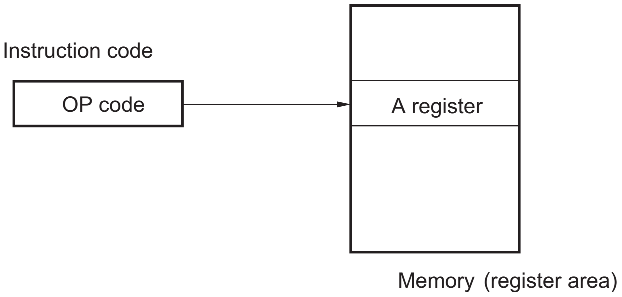
[Function]

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

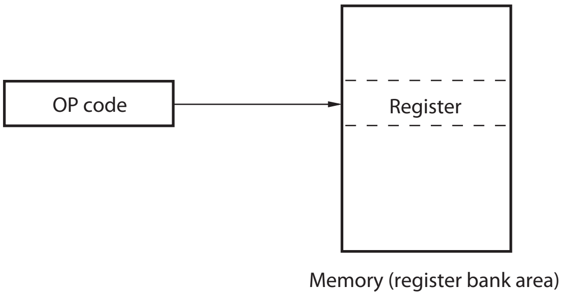
[Function]

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



<R> 3.4.3 Direct addressing

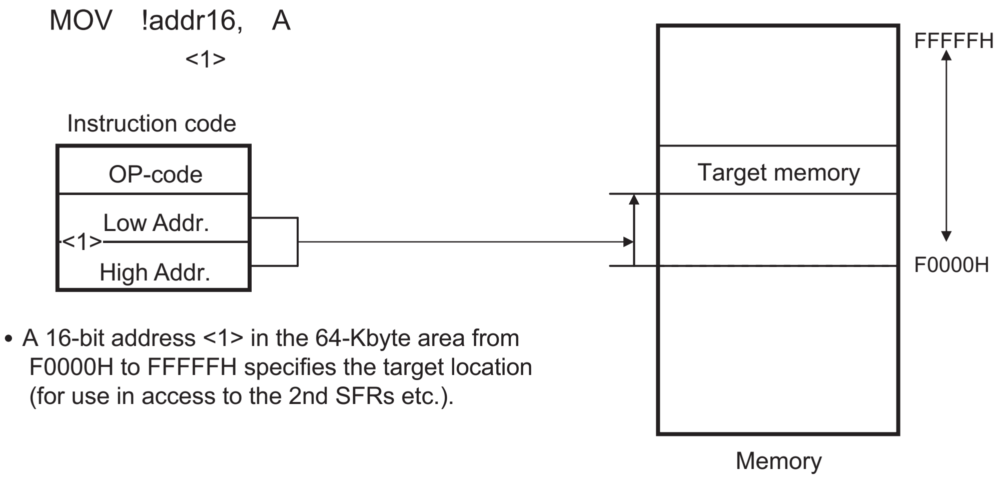
[Function]

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

[Operand format]

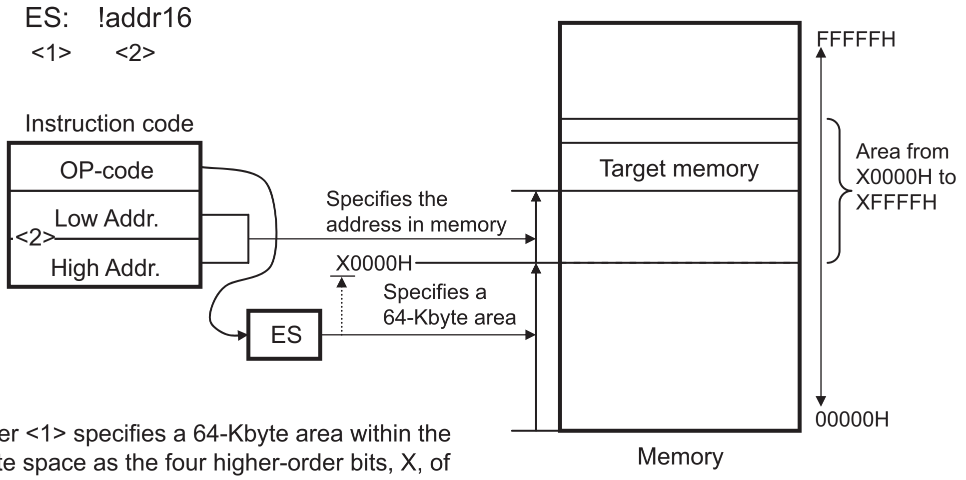
Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable: automatically added F of higher 4-bit addresses)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

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]

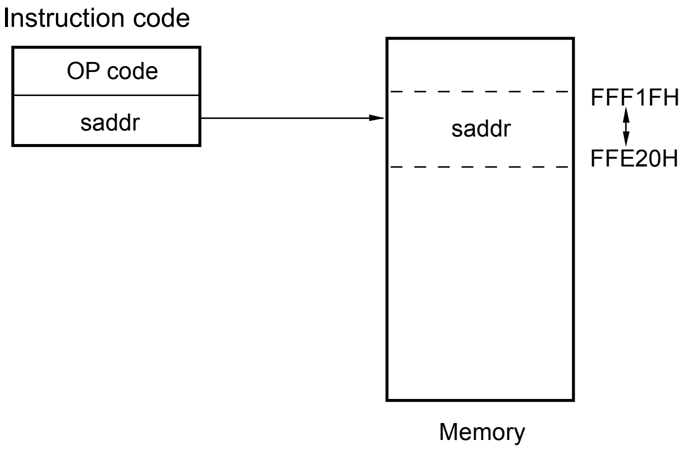
Short direct addressing directly specifies the target addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFE20H to FFF1FH.

Note that it is prohibited to use the area from FFEE0H to FFEF7H. In the products with 128 bytes of RAM, it is also prohibited to use the area from FFE20H to FFE5FH.

[Operand format]

Identifier	Description
SADDR	Label or FFE20H to FFF1FH immediate data
SADDRP	Label or FFE20H to FFF1FH immediate data (only even address 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

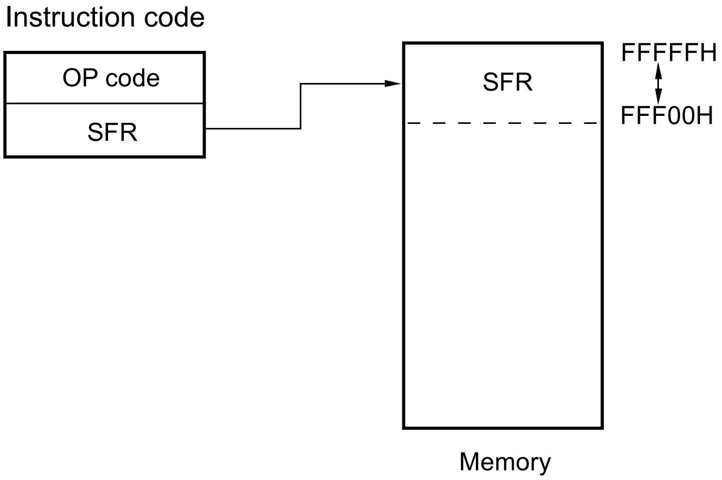
[Function]

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 only)

Figure 3-21. Outline of SFR Addressing



<R> 3.4.6 Register indirect addressing

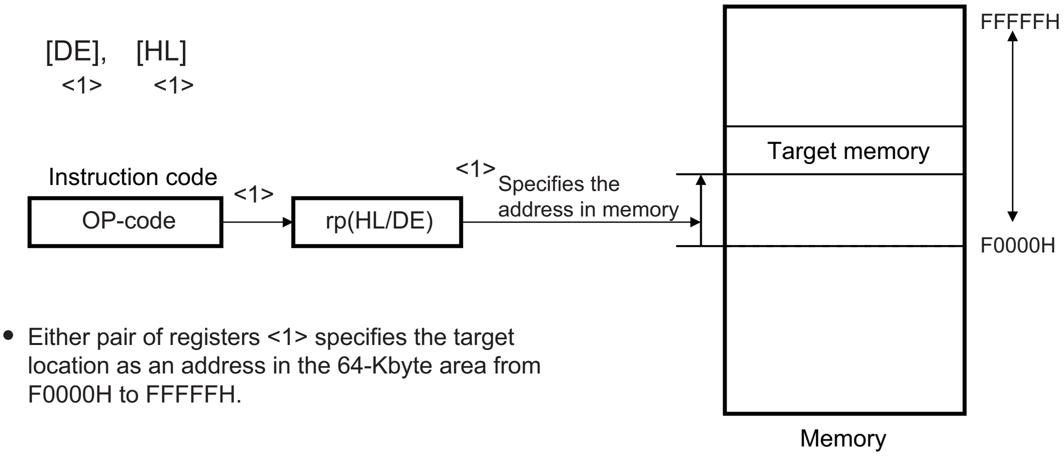
[Function]

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]

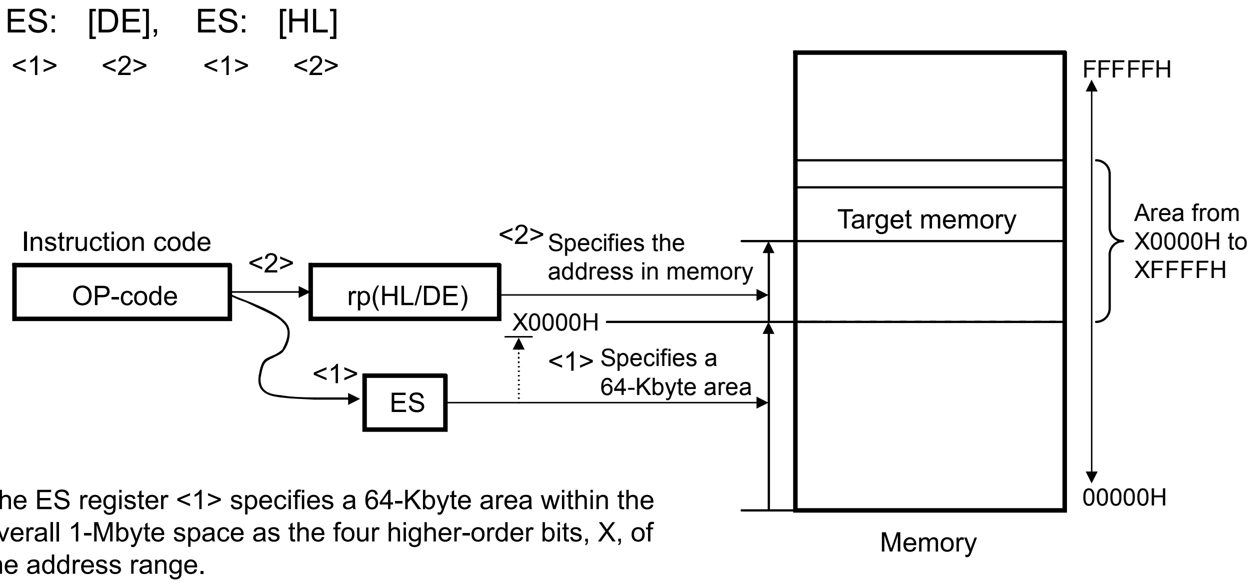
Identifier	Description
–	[DE], [HL] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

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.

<R> 3.4.7 Based addressing

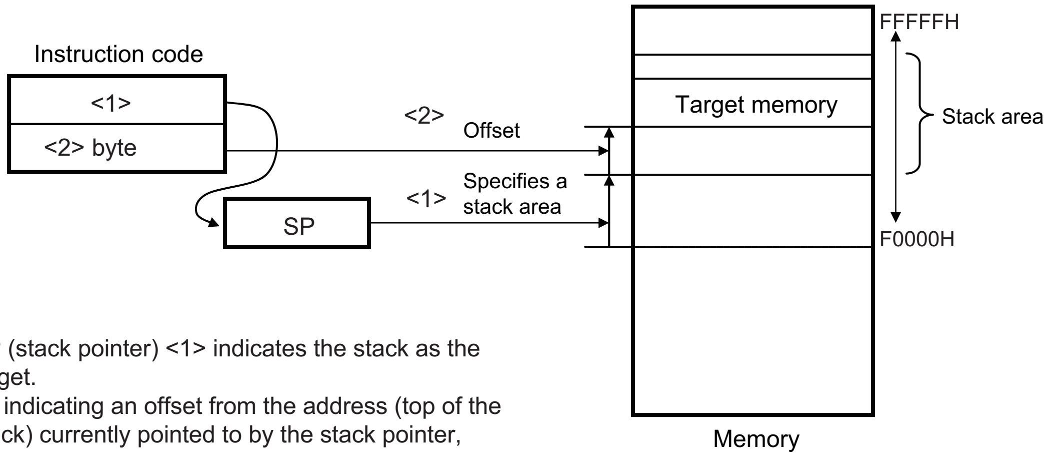
[Function]

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]

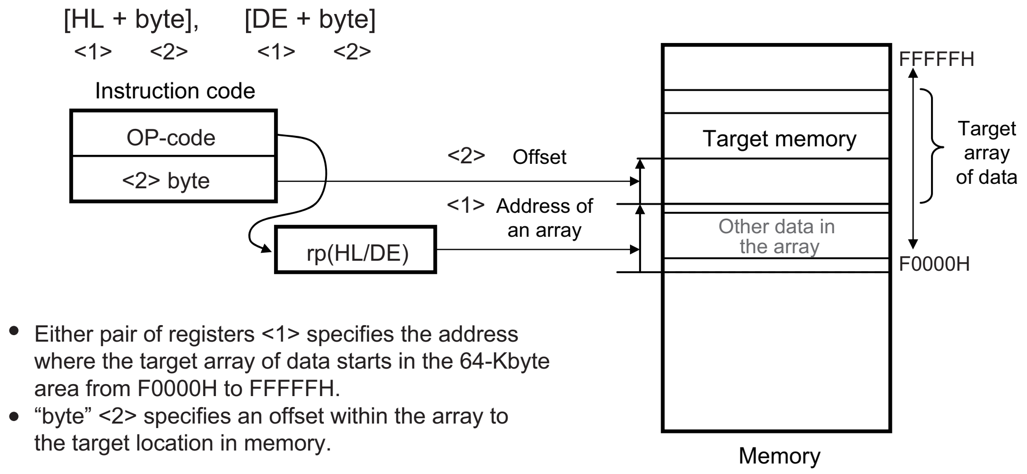
Identifier	Description
–	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
–	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
–	word[BC] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL + byte], ES:[DE + byte] (higher 4-bit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



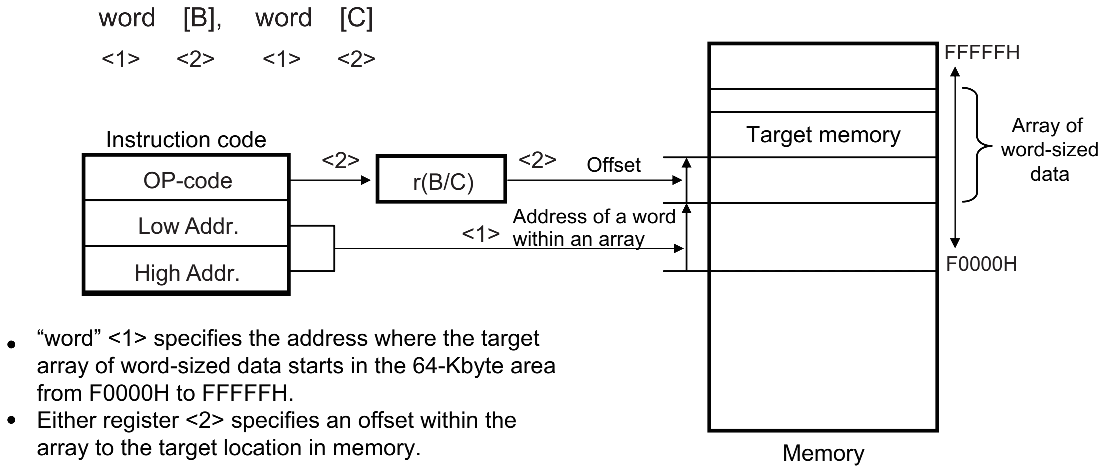
- SP (stack pointer) <1> indicates the stack as the target.
- By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, “byte” <2> indicates the target memory (SP + byte).

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



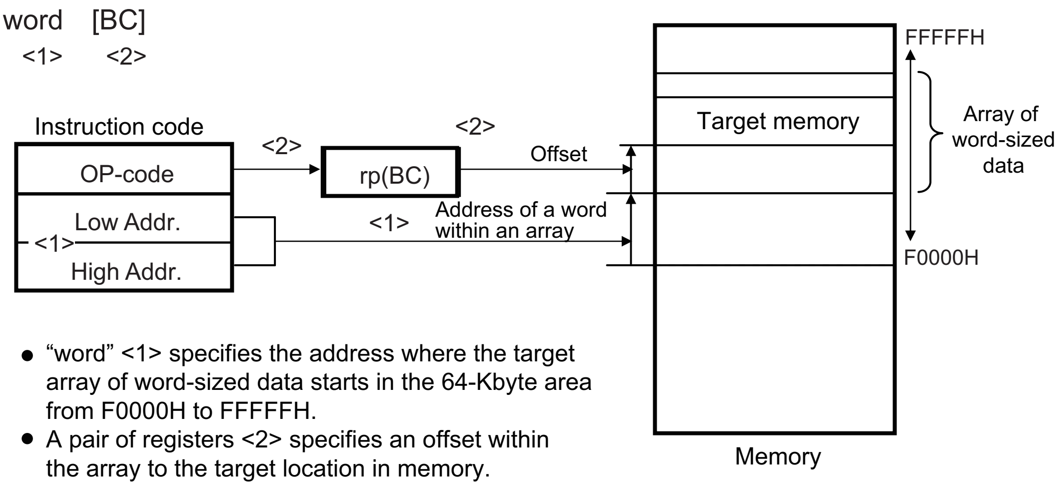
- Either pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- “byte” <2> specifies an offset within the array to the target location in memory.

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



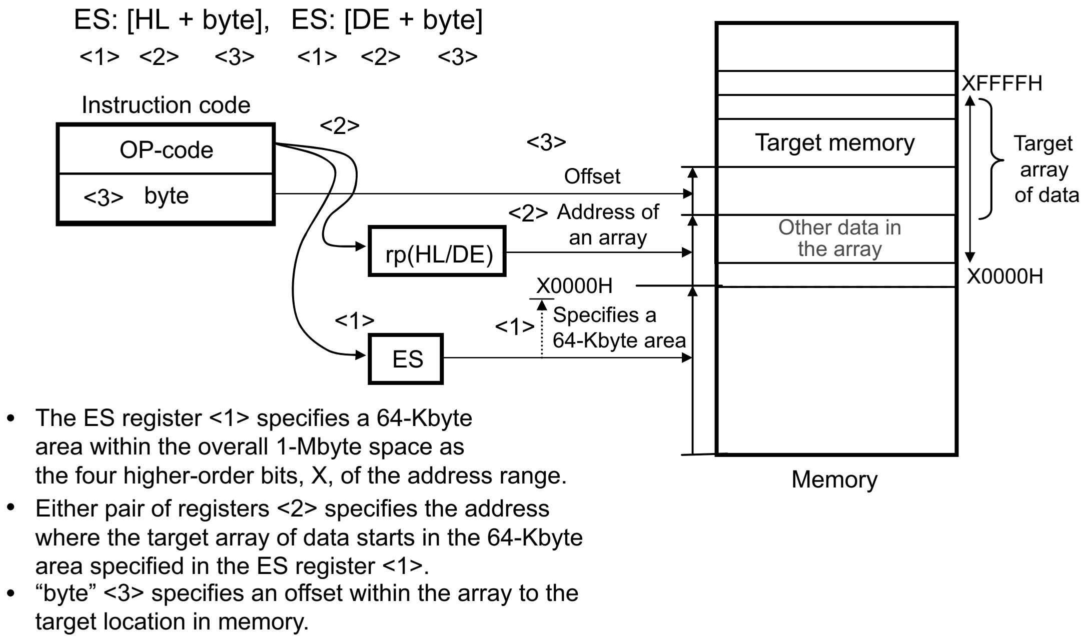
- “word” <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3-27. Example of word[BC]



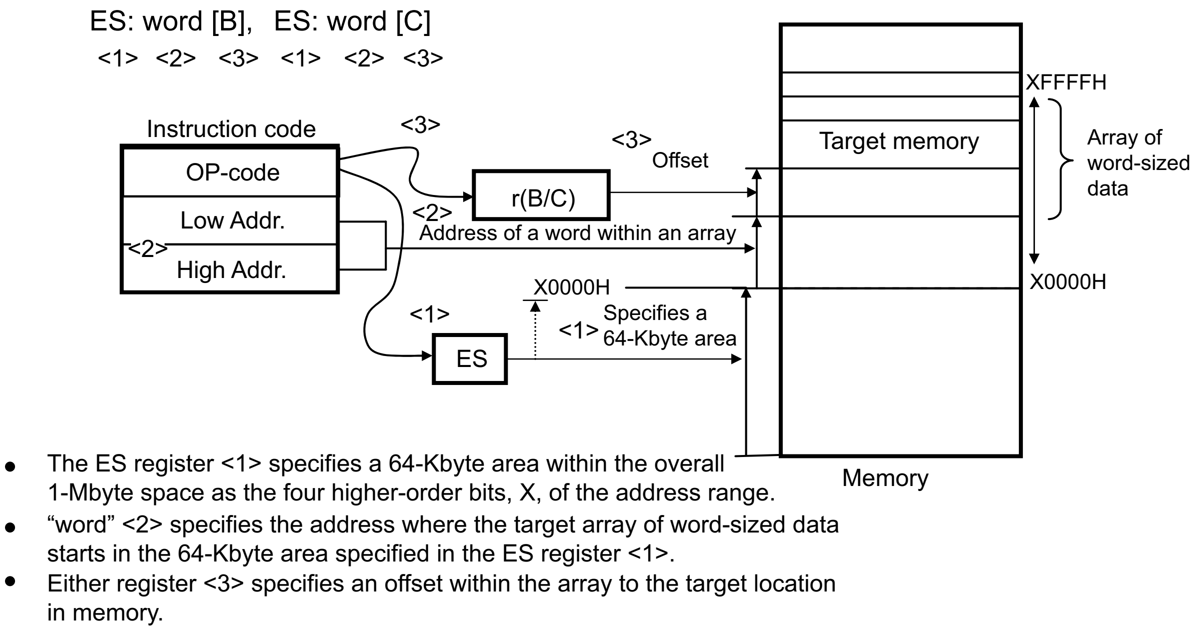
- “word” <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

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



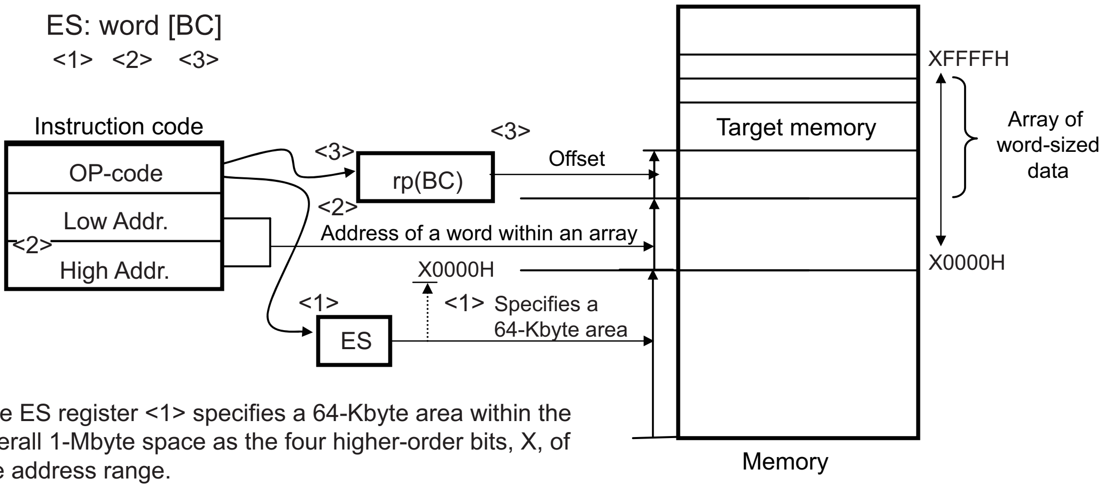
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- “byte” <3> specifies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

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



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

<R> 3.4.8 Based indexed addressing

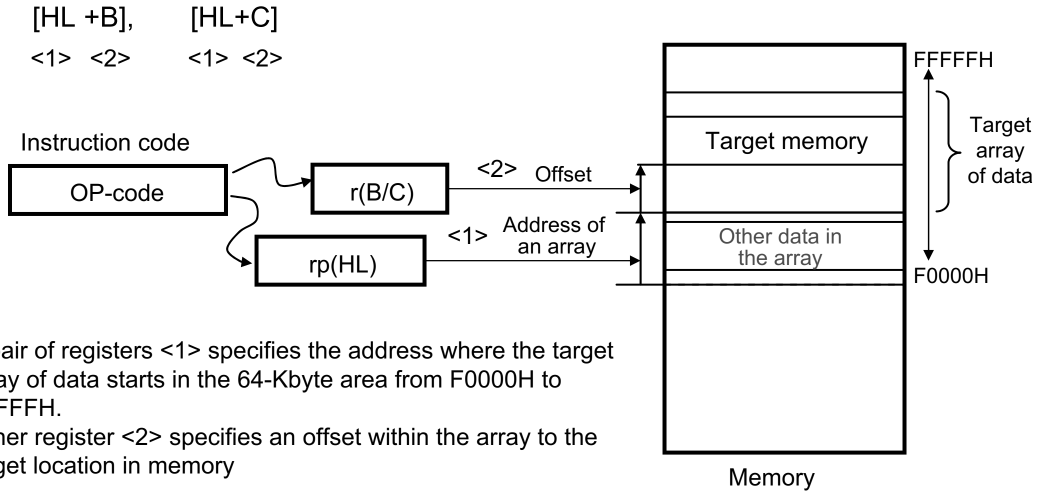
[Function]

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]

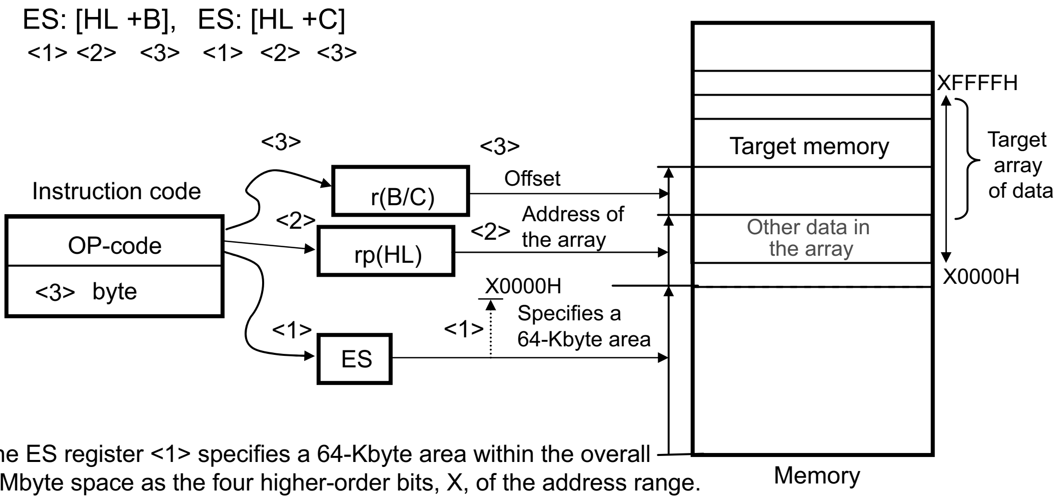
Identifier	Description
–	[HL+B], [HL+C] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL+B], ES:[HL+C] (higher 4-bit addresses are specified by the ES register)

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.

<R> 3.4.9 Stack addressing

[Function]

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

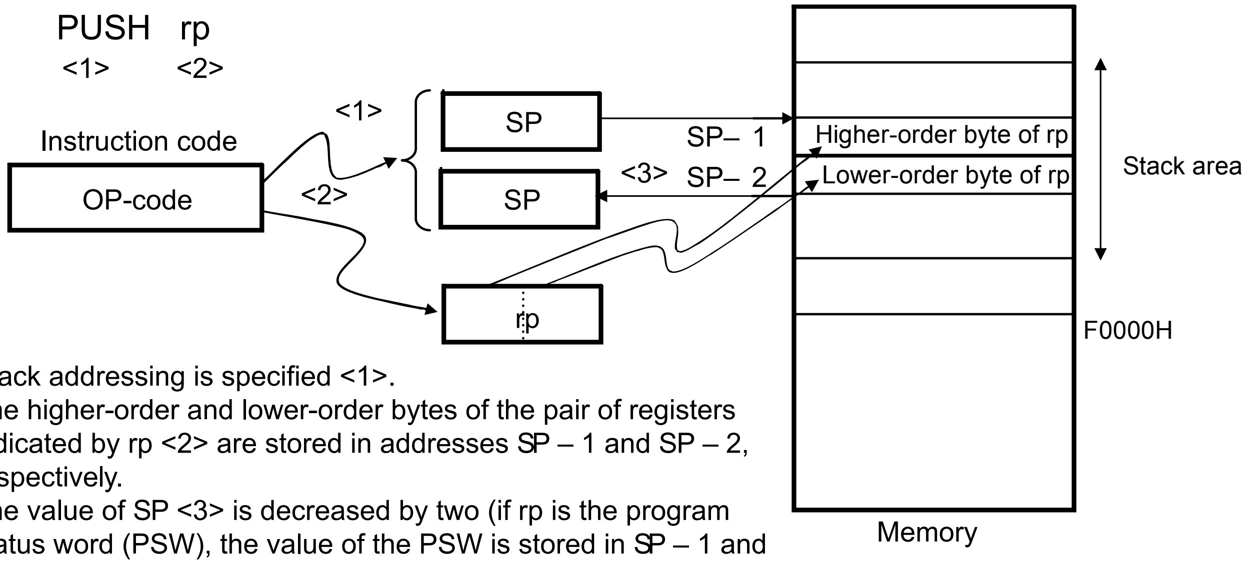
Only the internal RAM area can be set as the stack area.

[Operand format]

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

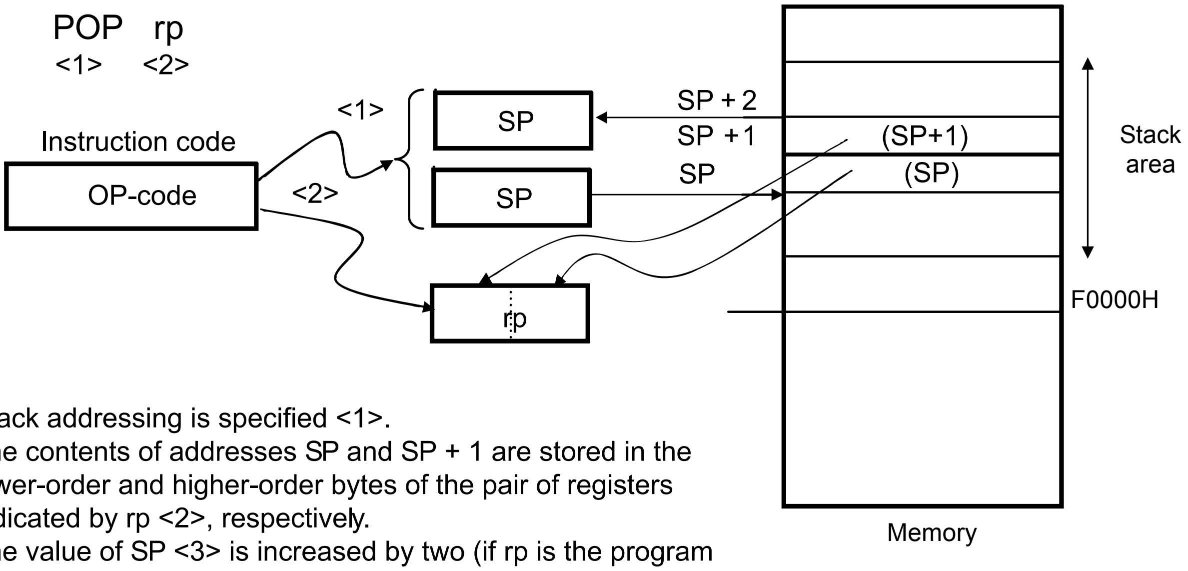
Each stack operation saves or restores data as shown in **Figures 3-33** to **3-38**.

Figure 3-33. Example of PUSH rp



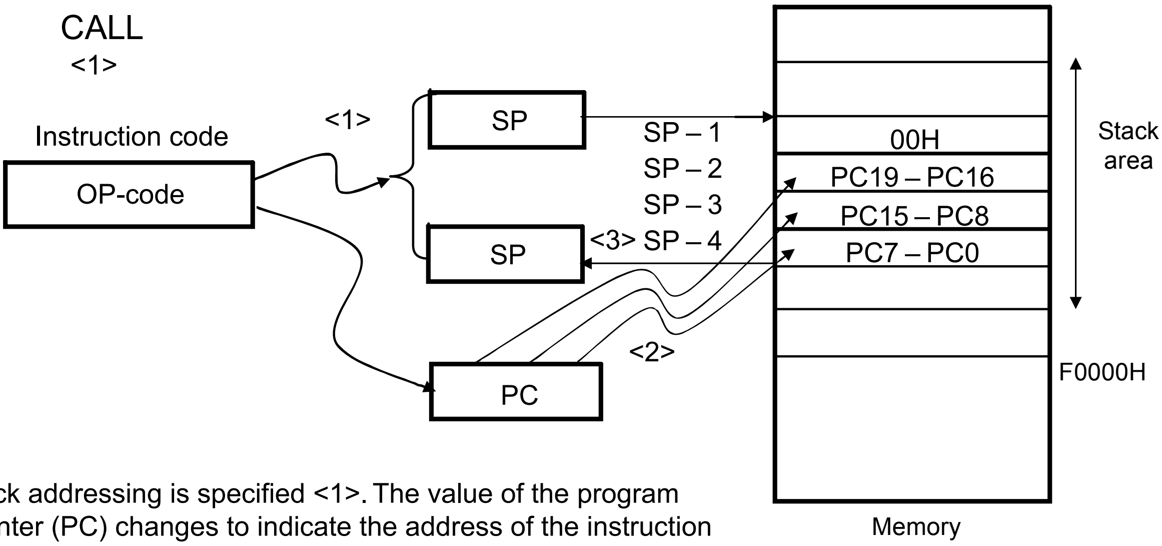
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP – 1 and SP – 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP –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

Figure 3-34. Example of POP



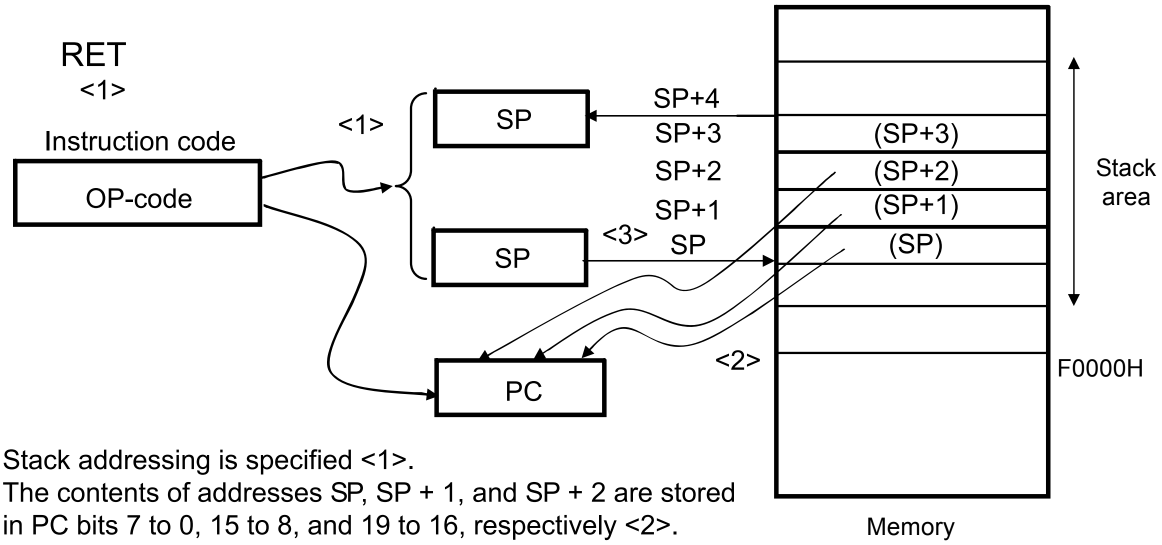
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP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

Figure 3-35. Example of CALL, CALLT



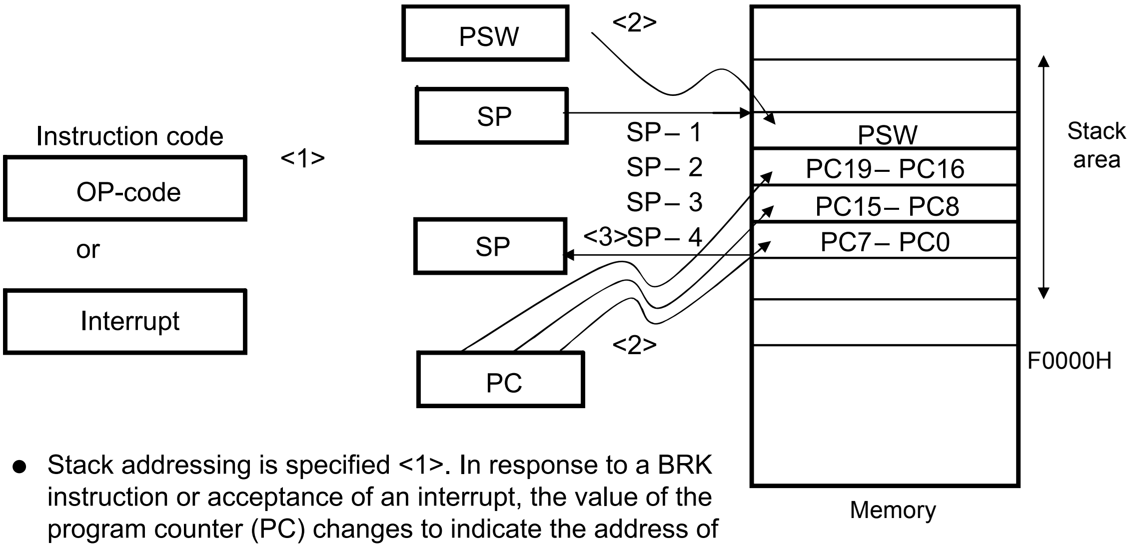
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP – 1, SP – 2, SP – 3, and SP – 4, respectively <2>.
- The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

Figure 3-36. Example of RET



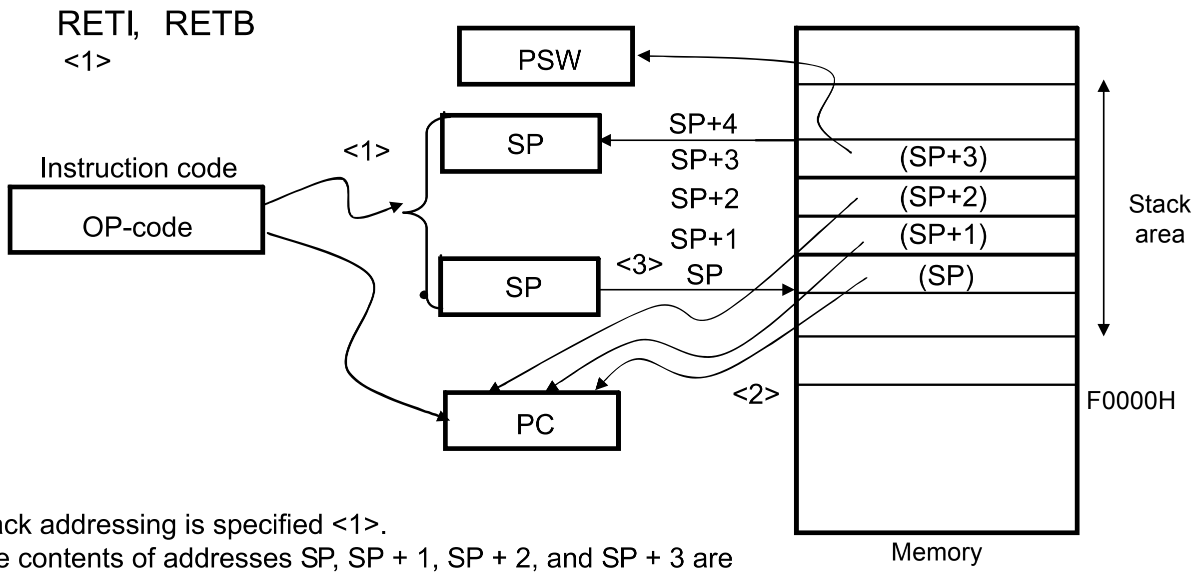
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

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 R7F0C80112ESP, R7F0C80212ESP 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 0, 4 (PM0, PM4) Port registers 0, 4, 12, 13 (P0, P4, P12, P13) Pull-up resistor option registers 0, 4, 12 (PU0, PU4, PU12) Port output mode register 0 (POM0) Port mode control register 0 (PMC0) Peripheral I/O redirection register (PIOR)
Port	Total: 8 (CMOS I/O: 6 (N-ch open-drain output (V_{DD} tolerance): 1), CMOS input: 2)
Pull-up resistor	Total: 7

4.2.1 Port 0

Port 0 is an I/O port with output latches. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P04 pins are used as input pins, use of the on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

<R> Output from the P00 pin can be specified as N-ch open-drain (V_{DD} tolerant) in 1-bit units using port output mode register 0 (POM0).

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

Reset signal generation sets port 0 to input mode.

4.2.2 Port 4

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

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

4.2.3 Port 12

Port 12 is an input port. Use of an on-chip pull-up resistor can be specified for P125 using pull-up resistor option register 12 (PU12) (valid after $\overline{\text{RESET}}$ input)[Note].

This port can also be used for key return input and reset input.

Note Once the power is turned on, P125 functions as the $\overline{\text{RESET}}$ input. The PORTSELB bit of the option byte (000C1H) defines whether this port operates as P125/KR1 or $\overline{\text{RESET}}$. When this pin is set to P125/KR1, do not input the low level to this pin during a reset by the selectable power-on-reset (SPOR) circuit and during the period from release from the reset by the SPOR circuit to the start of normal operation. If the low level is input during this period, the chip will remain in the reset state in response to the external reset. Accordingly, the pull-up resistor is enabled after the power is turned on.

4.2.4 Port 13

Port 13 is an input port.

This port can also be used for timer input and external interrupt input.

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- Port mode registers 0, 4 (PM0, PM4)
- Port registers 0, 4, 12, 13(P0, P4, P12, P13)
- Pull-up resistor option registers 0, 4, 12(PU0, PU4, PU12)
- Port output mode register 0 (POM0)
- Port mode control register 0 (PMC0)
- Peripheral I/O redirection register (PIOR)

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

Table 4-2. PMx, Pxx, PUxx, POMx, PMCxx Registers and the Bits

Port		Bit Name				
		Px Register	PMx Register	PUx Register	POMx Register	PMCx Register
PORT0	0	P00	PM00	PU00	POM00	–
	1	P01	PM01	PU01	–	PMC01
	2	P02	PM02	PU02	–	PMC02
	3	P03	PM03	PU03	–	PMC03
	4	P04	PM04	PU04	–	PMC04
PORT4	0	P40	PM40	PU40	–	–
PORT12	5	P125	–	PU125	–	–
PORT13	7	P137	–	–	–	–

The format of each register is described below.

4.3.1 Port mode registers 0, 4(PM0, PM4)

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 an Alternate Function Is Used**.

Figure 4-1. Format of Port Mode Registers 0, 4 (PM0, PM4)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W

PMmn	Pmn pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

m = 0, 4; n = 0 to 4

4.3.2 Port registers 0, 4, 12, 13 (P0, P4, P12, P13)

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 sets the P12 and P13 registers to the undefined value, and clears the other registers to 00H.

Note In the ports that 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.
In addition, in the output latch that are set up as $\overline{\text{RESET}}$ pin for P125, 1 is always read.

Figure 4-2. Format of Port Register 0, 4, 12, 13 (P0, P4, P12, P13)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	P04	P03	P02	P01	P00	FFF00H	00H (output latch)	R/W
P4	0	0	0	0	0	0	0	P40	FFF04H	00H (output latch)	R/W
P12	0	0	P125	0	0	0	0	0	FFF0CH	Undefined	R
P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R

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

m = 0, 4, 12, 13; n = 0 to 5, 7

4.3.3 Pull-up resistor option registers 0, 4, 12 (PU0, PU4, PU12)

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 satisfied following three conditions which the use of an on-chip pull-up resistor has been specified in these registers.

- PMmn = 1 (Input mode)
- Sets the digital input of PMCmn register
- POM0n = 0

On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of these registers.

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

Reset signal generation sets PU4 to 01H, PU12 to 20H, and clears PU0 to 00H.

Figure 4-3. Format of Pull-up Resistor Option Registers 0, 4, 12 (PU0, PU4, PU12)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W
PU12	0	0	PU125 Note	0	0	0	0	0	F003CH	20H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Note This bit can be only manipulated when the P125/KR1 function is selected (PORTSELB = 0).

Remark m = 0, 4, 12; n = 0 to 5

4.3.4 Port output mode register (POM0)

These registers set CMOS output or N-ch open drain output 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 00H.

Figure 4-4. Format of Port Output Mode Register 0 (POM0)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	0	0	0	0	POM00	F0050H	00H	R/W

POMmn	Pmn pin output mode selection
0	When this bit is set to 0 in the output mode, normal output mode is set. In the input mode, the setting of the PUmn bit is enabled.
1	When this bit is set to 1 in the output mode, N-ch open-drain output (V_{DD} toleranct) mode is set. In the input mode, the setting of the PUmn bit is disabled.

m = 0; n = 0

4.3.5 Port mode control registers (PMC0)

These registers set the digital I/O or analog input 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.

Figure 4-5. Format of Port Mode Control Register 0 (PMC0)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	1	1	1	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W

PMCmn	Pmn pin digital I/O/analog input selection
0	Digital I/O (alternate function other than analog input)
1	Analog input

m = 0; n = 1 to 4

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

4.3.6 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 clears this register to 00H.

Figure 4-6. Format of Peripheral I/O Redirection Register (PIOR)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIOR	0	0	0	0	0	PIOR2	PIOR1	PIOR0	F0077H	00H	R/W

Bit	Function	Setting value	
		0	1
PIOR2	INTP1	P00	P03
PIOR1	TI01/TO01	P04	P40
PIOR0	PCLBUZ0	P02	P40

<R> **Caution It is prohibited to set PIOR0 and PIOR1 to 1 at the same time.**

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.

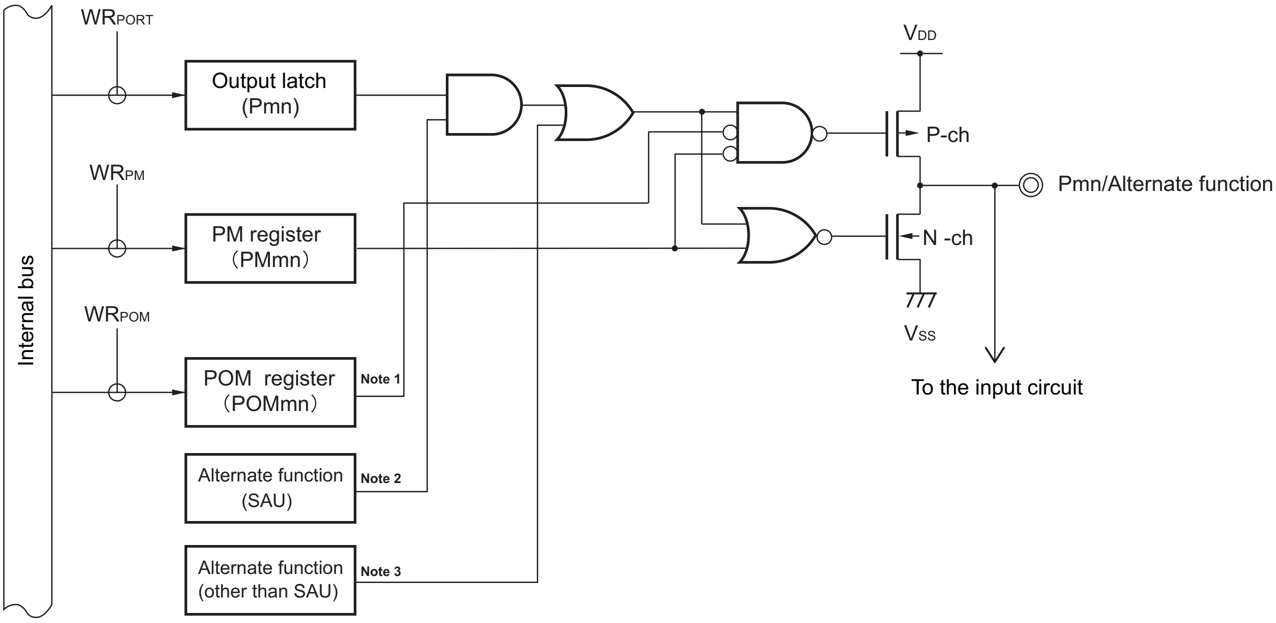
<R> 4.5 Register Settings When an Alternate Function Is Used

4.5.1 Basic concepts on using an alternate function

If a given pin is also used alternately for analog input, first in the port mode control register 0 (PMC0) specify whether the pin is to be used in analog input or digital output.

The basic configuration of an output circuit for pins that are used in digital I/O is shown in Figure 4-7. The output from the SAU function doubling as an output from the port output latch is input into the AND gate. The output from the AND gate is input into the OR gate. To the other input pin for the OR gate, the outputs from alternate non-SAU functions (TAU, clock/buzzer output, etc.) are connected. When such a pin is used as a port or alternate function, the alternate function that is not used must not interfere with the output from the function to be used. Table 4-3 summarizes underling concepts of specifying basic settings for making that distinction.

Figure 4-7. Basic Configuration of the Output Circuit for the Pins



Notes 1. In the absence of a POM register, this signal should be considered Low (0).

2. In the absence of an alternate function, this signal should be considered High (1).

3. In the absence of an alternate function, this signal should be considered Low (0).

Remark m: port number (m = 0, 4, 12, 13); n: bit number (n = 0 to 7)

Table 4-3. Basic Settings

Output function of the pin used	Output settings for alternate functions that are not used		
	Port function	SAU output function	Non-SAU output function
Port output function	–	Output: High (1)	Output: Low (0)
SAU output function	High(1)	–	Output: Low (0)
Non-SAU output function	Low(0)	Output: High (1)	Output: Low (0) Note

Note Since more than one non-SAU output function can be assigned to a given pin, the output from an alternate function that is not used must be set to Low (0). For specific settings methods, see **4.5.2 Register settings for alternate functions that do not use an output function**.

4.5.2 Register settings for alternate functions that do not use an output function

If the output from an alternate function associated with a pin is not used, the settings described below must be specified. If the pin is subject to a peripheral I/O redirect function, the output can be changed to another pin by setting the peripheral I/O redirection register (PIOR). In this manner, the port function or another alternate function that is assigned to the target pin can be used.

(1) SOp = 1/TxDq = 1 (when not using the serial output (SOp/TxDq) of SAU)

In situations where serial output (SOp/TxDq) is not used, such as when SAU is used exclusively for serial input, set the bits in the serial output enable register 0 (SOE0) associated with the output that is not used to 0 (output disabled), and the SO0n bit in the serial output register 0 (SO0) to 1 (High). This is the same as the default settings.

(2) SCKp = 1/SDAr = 1/SCLr = 1 (when not using the channel n of SAU)

When not using SAU, set the bit n (SE0n) in the serial channel enable status register 0 (SE0) to 0 (operation halted status), set the bits in the serial output enable register 0 (SOE0) associated with the output that is not used to 0 (output disabled), and the SO0n and CKO0n bits in the serial output register 0 (SO0) to 1 (High). This is the same as the default settings.

(3) TO0n = 0 (when not using the output from the channel n of TAU)

When not using the TO0n output from TAU, set the bits in the timer output enable register 0 (TOE0) associated with the output that is not used to 0 (output disabled), and the bits in the timer output register 0 (TO0) to 0 (Low). This is the same as the default settings.

(4) PCLBUZ0 = 0 (when not using the clock output/buzzer output)

When not using the clock output/buzzer output, set the PCLOE0 bit in the clock output selection register 0 (CKS0) to 0 (output disabled). This is the same as the default settings.

4.5.3 Example of register settings for port and alternate functions used

Table 4-5 shows examples of register settings for port and alternate functions that are used. Registers that control the port functions should be set as indicated in Table 4-4. For conventions used in Table 4-4, see the remarks provided below:

Remarks

–:	Excluded
×:	don’t care
PIOR:	Peripheral I/O redirection register
POM0:	Port output mode register 0
PMC0:	Port mode control register 0
PMn:	Port mode register n (n = 0, 4)
Pm:	Port output latch (m = 0, 4, 12, 13)

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

<R>

Table 4-5. Examples of Register And Output Latch Settings With Pin Functions (1/2)

Pin	Function		PIOR	POM0	PMC0	PMn	Pm	Alternate function output	
	Name	I/O						SAU output function	Non-SAU
P00	P00	Input	–	×	–	1	×	×	–
		Output	–	0	–	0	0/1	TxD0/SO00 = 1	–
		N-ch open-drain output	–	1	–	0	0/1		
	SO00	Output	–	0	–	0	1	×	–
	TXD0	Output	–	0/1	–	0	1	×	–
	INTP1	Input	PIOR2 = 0	×	–	1	×	×	–
P01	P01	Input	–	×	0	1	×	×	–
		Output	–	0	0	0	0/1	×	–
	ANI0	Analog input	–	×	1	1	×	×	–
	SI00	Input	–	×	0	1	×	×	–
	RxD0	Input	–	×	0	1	×	×	–
	KR2	Input	–	×	0	1	×	×	–
P02	P02	Input	–	–	0	1	×	×	×
		Output	–	–	0	0	0/1	SCK00 = 1	PCLBUZ0 = 0
	ANI1	Analog input	–	–	1	1	×	×	×
	SCK00	Input	–	–	0	1	×	×	×
		Output	–	–	0	0	1	×	PCLBUZ0 = 0
	PCLBUZ0	Output	PIOR0 = 0	–	0	0	0	SCK00 = 1	×
	KR3	Input	–	–	0	1	×	×	×
P03	P03	Input	–	–	0	1	×	–	×
		Output	–	–	0	0	0/1	–	TO00 = 0
	ANI2	Analog input	–	–	1	1	×	–	×
	TO00	Output	–	–	0	0	0	–	×
	KR4	Input	–	–	0	1	×	–	×
	(INTP1)	Input	PIOR2 = 1	–	0	1	×	–	×
P04	P04	Input	–	–	0	1	×	–	×
		Output	–	–	0	0	0/1	–	TO01 = 0
	ANI3	Analog input	–	–	1	1	×	–	×
	TI01	Input	PIOR1 = 0	–	0	1	×	–	×
	TO01	Output	PIOR1 = 0	–	0	0	0	–	×
	KR5	Input	–	–	0	1	×	–	×
P40	P40	Input	–	–	–	1	×	–	×
		Output	–	–	–	0	0/1	–	(PCLBUZ0) = 0 (TO01) = 0
	KR0	Input	–	–	–	1	×	–	×
	(PCLBUZ0)	Output	PIOR0 = 1	–	–	0	0	–	(TO01) = 0
	(TI01)	Input	PIOR1 = 1	–	–	1	×	–	×
	(TO01)	Output	PIOR1 = 1	–	–	0	0	–	(PCLBUZ0) = 0

Table 4-5. Examples of Register And Output Latch Settings With Pin Functions (2/2)

Pin	Function		PIOR	POM0	PMC0	PMn	Pm	Notes
	Name	I/O						
P125	P125	Input	–	–	–	–	×	Optional bytes 000C1H PORTSELB = 0
	KR1	Input	–	–	–	–	×	
	$\overline{\text{RESET}}$	Input	–	–	–	–	×	Optional bytes 000C1H PORTSELB = 1
P137	P137	Input	–	–	–	–	×	–
	TI00	Input	–	–	–	–	×	–
	INTP0	Input	–	–	–	–	×	–

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 P00 is an output port, P01 to P04 are input ports (all pin statuses are high level), and the port latch value of port 0 is 00H, if the output of output port P00 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 0 is FFH.

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

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

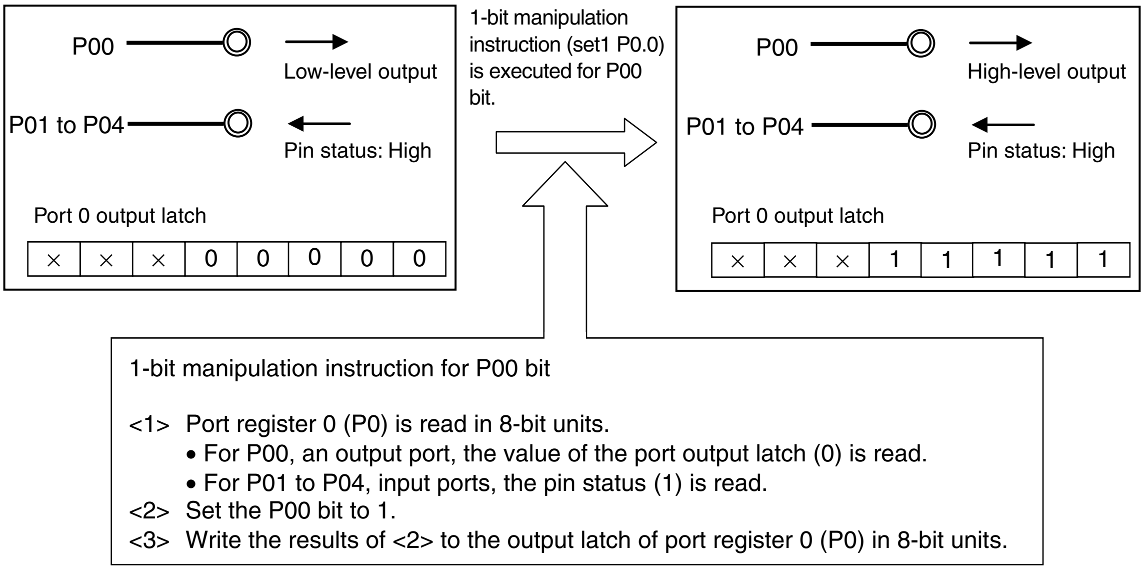
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P00, which is an output port, is read, while the pin statuses of P01 to P04, which are input ports, are read. If the pin statuses of P01 to P04 are high level at this time, the read value is EH.

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

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

Figure 4-8. Bit Manipulation Instruction (P00)



4.6.2 Notes on specifying the pin settings

For an output pin to which multiple functions are assigned and a given function is selected for output, any unused alternate function must be set to its initial state so as to prevent conflict with the selected function. 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 an Alternate Function Is Used**.

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 peripheral functions 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> High-speed on-chip oscillator

The frequency at which to oscillate can be selected from among f_{IH} = 20/10/5/2.5/1.25 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.
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 5-3 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)				
	1.25	2.5	5	10	20
$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	√	√	√	√	√
$2.4 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	√	√	√	–	–

<R> **Note** Use this product within the voltage range from 2.57 to 5.5 V because the detection voltage (V_{SPOR}) of the selectable power-on-reset (SPOR) circuit should also be considered.

(2) 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 watchdog timer runs on the low speed on-chip oscillator clock.

This clock operates when bit 4 (WDTON) of the option byte (000C0H) is set to 1.
However, when WDTON = 1 and bit 0 (WDSTBYON) of the option byte (000C0H) is 0, oscillation of the LOCO stops if the HALT or STOP instruction is executed.

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

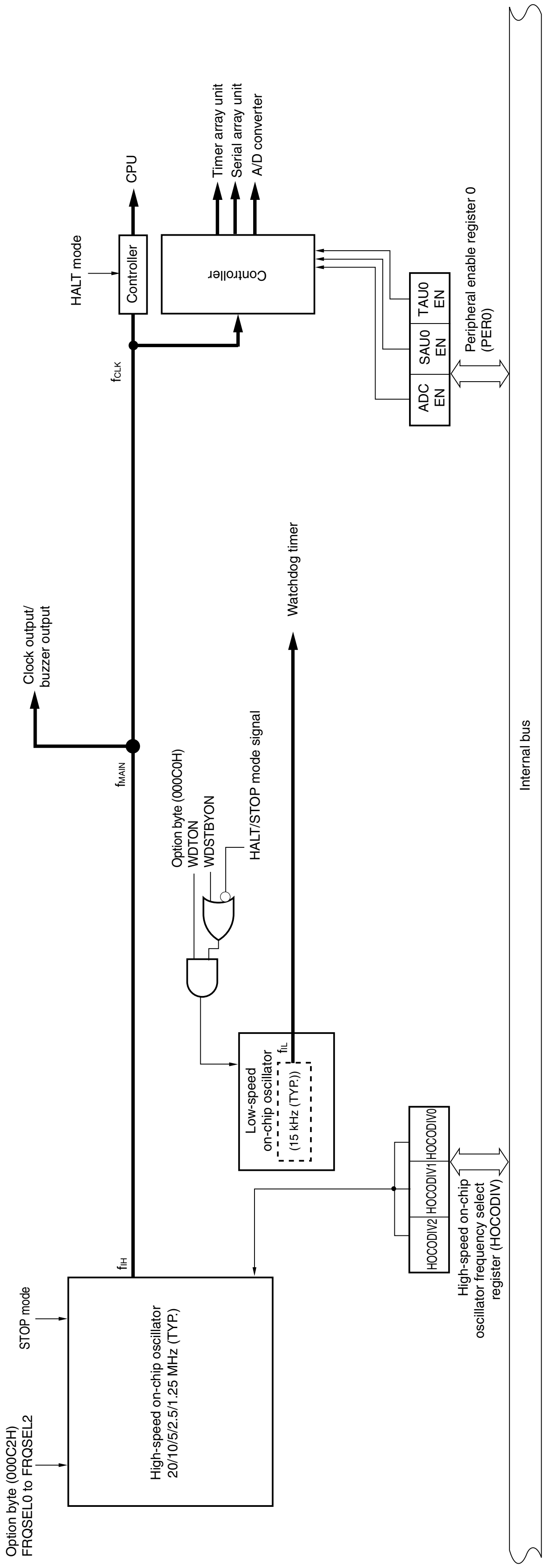
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Peripheral enable register 0 (PER0) High-speed on-chip oscillator frequency selection register (HOCODIV)
Oscillators	High-speed on-chip oscillator Low-speed on-chip oscillator

Figure 5-1. Block Diagram of Clock Generator



Remark	f_{IH}:	High-speed on-chip oscillator clock frequency
	f_{MAIN}:	Main system 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 registers are used to control the clock generator.

- Peripheral enable register 0 (PER0)
- High-speed on-chip oscillator frequency selection register (HOCODIV)

5.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to 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.

- A/D converter
- Serial array unit 0
- Timer array unit 0

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

Address: F00F0H After reset: 00H R/W

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

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

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

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

5.3.2 High-speed on-chip oscillator frequency selection register (HOCODIV)

This register is used to change the frequency of the high-speed on-chip oscillator clock set with the option byte (000C2H).

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

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

Figure 5-3. Format of High-Speed On-Chip Oscillator Frequency Selection Register (HOCODIV)

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

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV 2	HOCODIV 1	HOCODIV 0

HOCODIV 2	HOCODIV 1	HOCODIV 0	High-speed on-chip oscillator clock frequency selection
0	0	1	20 MHz
0	1	0	10 MHz
0	1	1	5 MHz
1	0	0	2.5 MHz
1	0	1	1.25 MHz
Other than above			Setting prohibited

Cautions 1. Set the HOCODIV register within the operable voltage range before and after the frequency change.

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

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

5.4 System Clock Oscillator

5.4.1 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the R7F0C80112, R7F0C80212. The frequency can be selected from among 20, 10, 5, 2.5, or 1.25 MHz by using the option byte (000C2H). The high-speed on-chip oscillator automatically starts oscillating after reset release.

5.4.2 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the R7F0C80112, R7F0C80212.

The low-speed on-chip oscillator clock is used only as the watchdog timer clock. The low-speed on-chip oscillator clock cannot be used as the CPU clock.

This clock operates when bit 4 (WDTON) of the option byte (000C0H) is set to 1.

When the watchdog timer is stopped, oscillation of the low-speed on-chip oscillator stops.

5.5 Clock Generator Operation

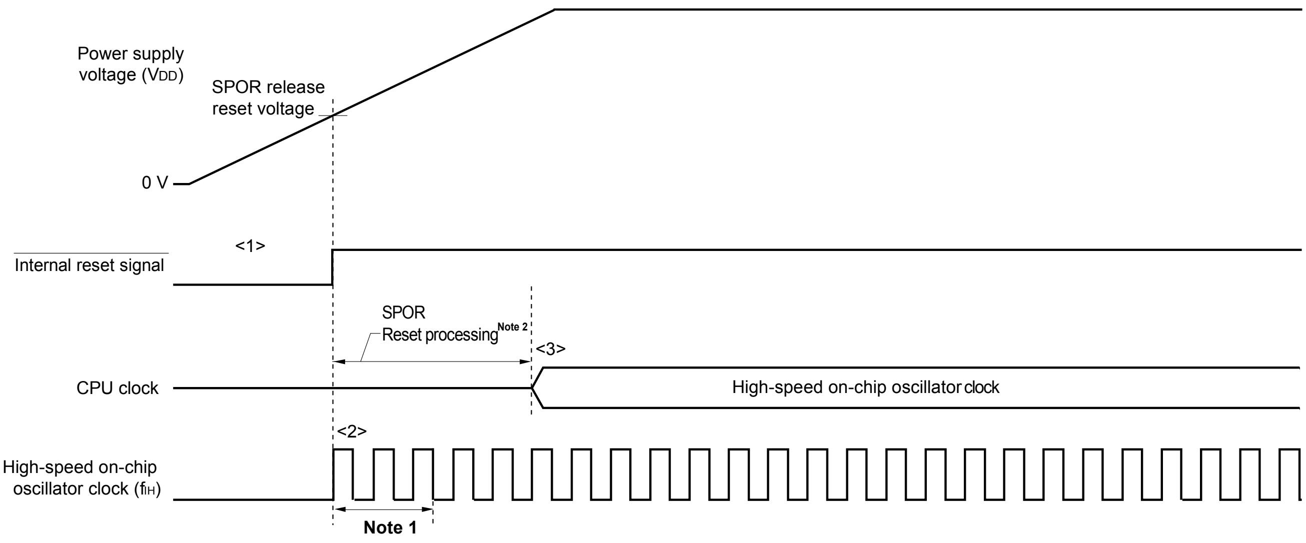
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 5-1**).

- Main system clock f_{MAIN}
- High-speed on-chip oscillator clock f_{IH}
- Low-speed on-chip oscillator clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the R7F0C80112, R7F0C80212.

When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-4.

Figure 5-4. 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 selectable power-on-reset (SPOR) circuit.

<2> When the power supply voltage exceeds detection voltage of the SPOR circuit, the reset is released and the high-speed on-chip oscillator automatically starts oscillation.

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

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

<R> **2.** For SPOR reset processing time, see **CHAPTER 15 SELECTABLE POWER-ON-RESET CIRCUIT**.

<R> 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 by using FRQSEL0 to FRQSEL2 of the option byte (000C2H). This frequency can be changed with the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

	7	6	5	4	3	2	1	0
Option byte (000C2H)	1	1	1	0	1	FRQSEL2	FRQSEL1	FRQSEL0

FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
0	0	1	20 MHz
0	1	0	10 MHz
0	1	1	5 MHz
1	0	0	2.5 MHz
1	0	1	1.25 MHz
Other than above			Setting prohibited

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

Address: F00A8H

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

HOCODIV 2	HOCODIV 1	HOCODIV 0	Selected frequency
0	0	1	20 MHz
0	1	0	10 MHz
0	1	1	5 MHz
1	0	0	2.5 MHz
1	0	1	1.25 MHz
Other than above			Setting prohibited

<R> **Cautions 1. Set the HOCODIV register within the operable voltage range before and after the frequency change.**

<R> **2. After the frequency is changed with the HOCODIV register, the frequency is switched after the following transition time has elapsed.**

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

5.6.2 CPU clock status transition diagram

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

Figure 5-5. CPU Clock Status Transition Diagram

<R>

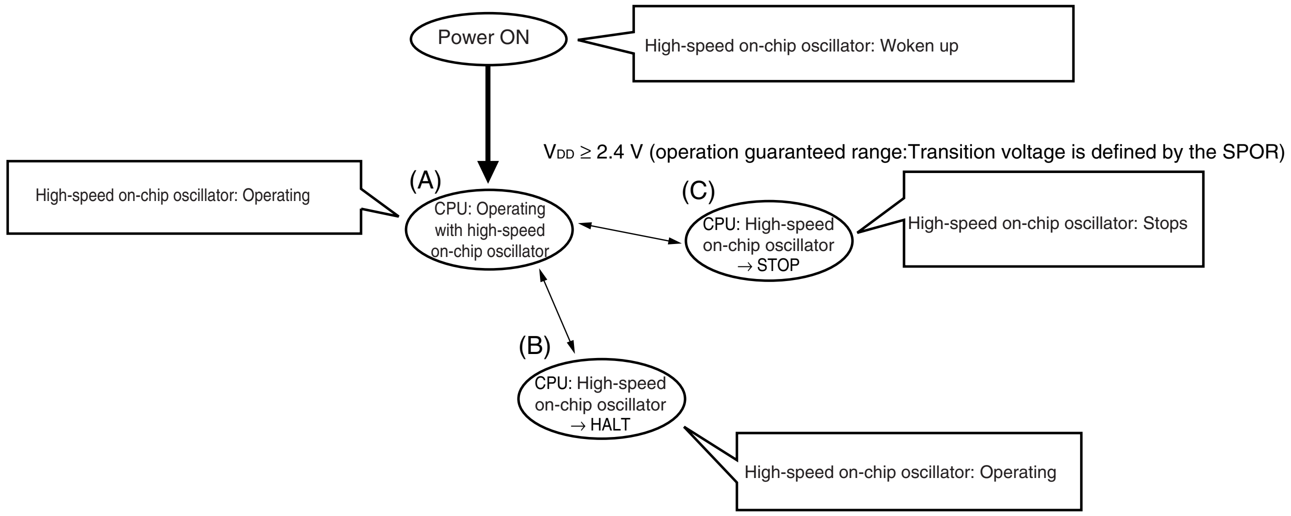


Table 5-2 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-2. CPU Clock Transition and SFR Register Setting Examples (1/3)

(1) • HALT mode (B) set while CPU is operating with high-speed on-chip oscillator clock (A)

Status Transition	Setting
(A) → (B)	Executing HALT instruction

(2) • STOP mode (C) set while CPU is operating with high-speed on-chip oscillator clock (A)

(Setting sequence) →

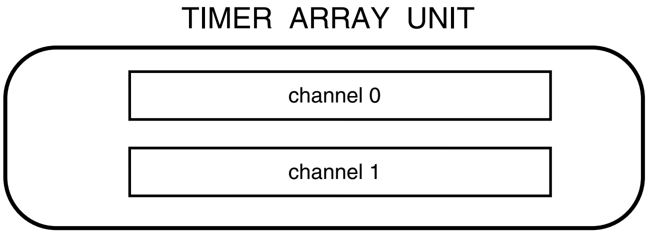
Status Transition	Setting	
(A) → (C)	Stopping peripheral functions that cannot operate in STOP mode	Executing STOP instruction

Remark (A) to (C) in Table 5-2 correspond to (A) to (C) in Figure 5-5.

CHAPTER 6 TIMER ARRAY UNIT

The timer array unit has two 16-bit timers.

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



For details about each function, see the table below.

Independent Channel Operation Function	Simultaneous Channel Operation Function
• Interval timer (→ refer to **6.7.1**) • Square wave output (→ refer to **6.7.1**) • External event counter (→ refer to **6.7.2**) • Divider^{Note} (→ refer to **6.7.3**) • Input pulse interval measurement (→ refer to **6.7.4**) • Measurement of high-/low-level width of input signal (→ refer to **6.7.5**) • Delay counter (→ refer to **6.7.6)**	• One-shot pulse output (→ refer to **6.8.1**) • PWM output (→ refer to **6.8.2**)

It is possible to use the 16-bit timer of channel 1 as two 8-bit timers (higher and lower). The functions that can use channel 1 as 8-bit timers are as follows:

- Interval timer/square wave output
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)
- PWM output (lower 8-bit timer only)

Note Only channel 0

6.1 Functions of Timer Array Unit

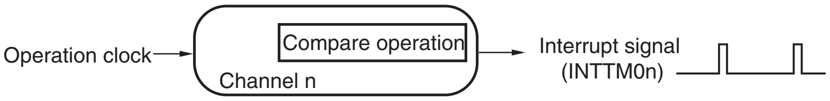
Timer array unit has the following functions.

6.1.1 Independent channel operation function

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

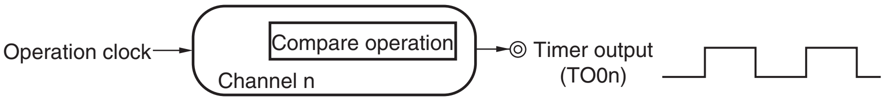
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTM0n) at fixed intervals.



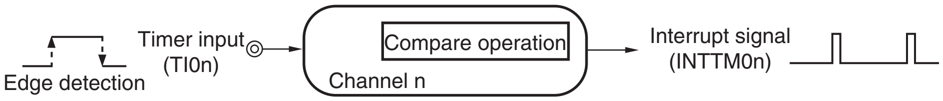
(2) Square wave output

A toggle operation is performed each time INTTM0n interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TO0n).



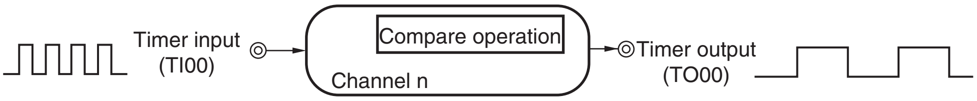
(3) External event counter

Each timer of a unit can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TI0n) has reached a specific value.



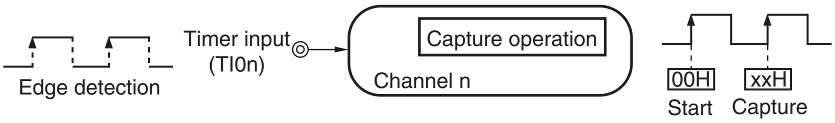
(4) Divider function

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



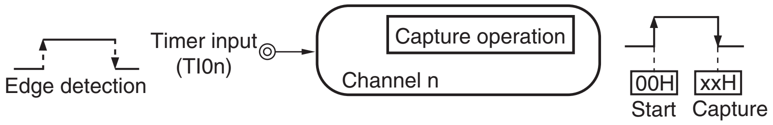
(5) Input pulse interval measurement

Counting is started by the valid edge of a pulse signal input to a timer input pin (TI0n). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



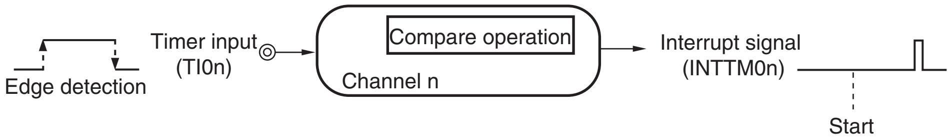
(6) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TI0n), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.



(7) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TI0n), and an interrupt is generated after any delay period.



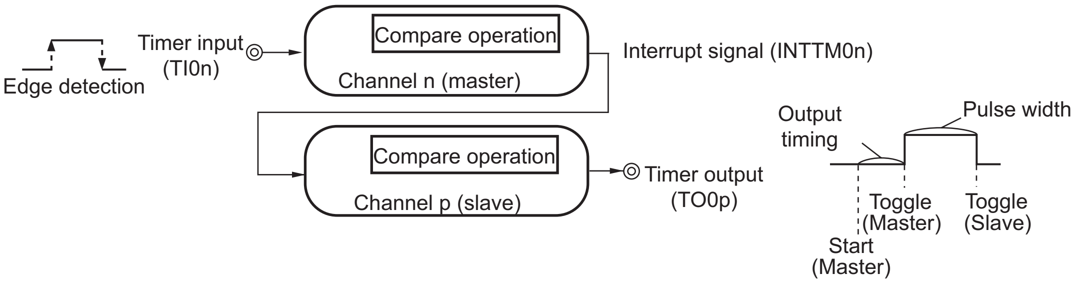
Remark n: Channel number (n = 0, 1)

6.1.2 Simultaneous channel operation function

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

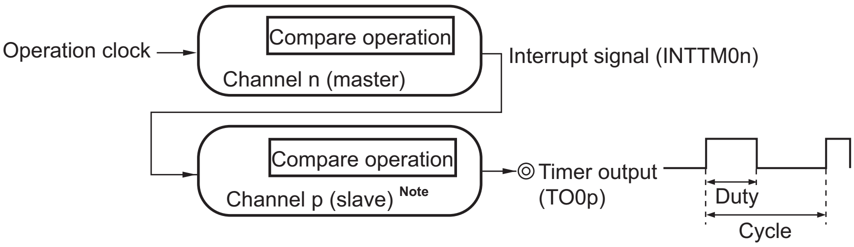
(1) One-shot pulse output

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.



Note This operation can be obtained with the eight lower bits of channel 1.

Caution There are several rules for using the simultaneous channel operation function.
For details, see 6.4.1 Basic Rules of Simultaneous Channel Operation Function.

Remark n: Channel number (n = 0, 1)
p: Slave channel number (p = 0)

6.1.3 8-bit timer operation function (channel 1 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 channel 1.

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 (Only Channel 1).

6.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 6-1. Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer/counter register 0n (TCR0nH, TCR0nL)
Register	Timer data register 0n (TDR0nH, TDR0nL)
Timer input	TI00, TI01
Timer output	TO00, TO01, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register 0 (TPS0) • Timer channel enable status register 0 (TE0, TEH0) • Timer channel start register 0 (TS0, TSH0) • Timer channel stop register 0 (TT0, TTH0) • Timer output enable register 0 (TOE0) • Timer output register 0 (TO0) • Timer output level register 0 (TOL0) • Timer output mode register 0 (TOM0)
	<Registers of each channel> • Timer mode register 0n (TMR0nH, TMR0nL) • Timer status register 0n (TSR0n) • Noise filter enable register 1 (NFEN1) • Port mode control register 0 (PMC0) • Port mode registers 0, 4 (PM0, PM4) • Port registers 0, 4 (P0, P4)

Remark n: Channel number (n = 0, 1).

Figures 6-1 and 6-2 show the block diagrams of the timer array unit.

Figure 6-1. Entire Configuration of Timer Array Unit

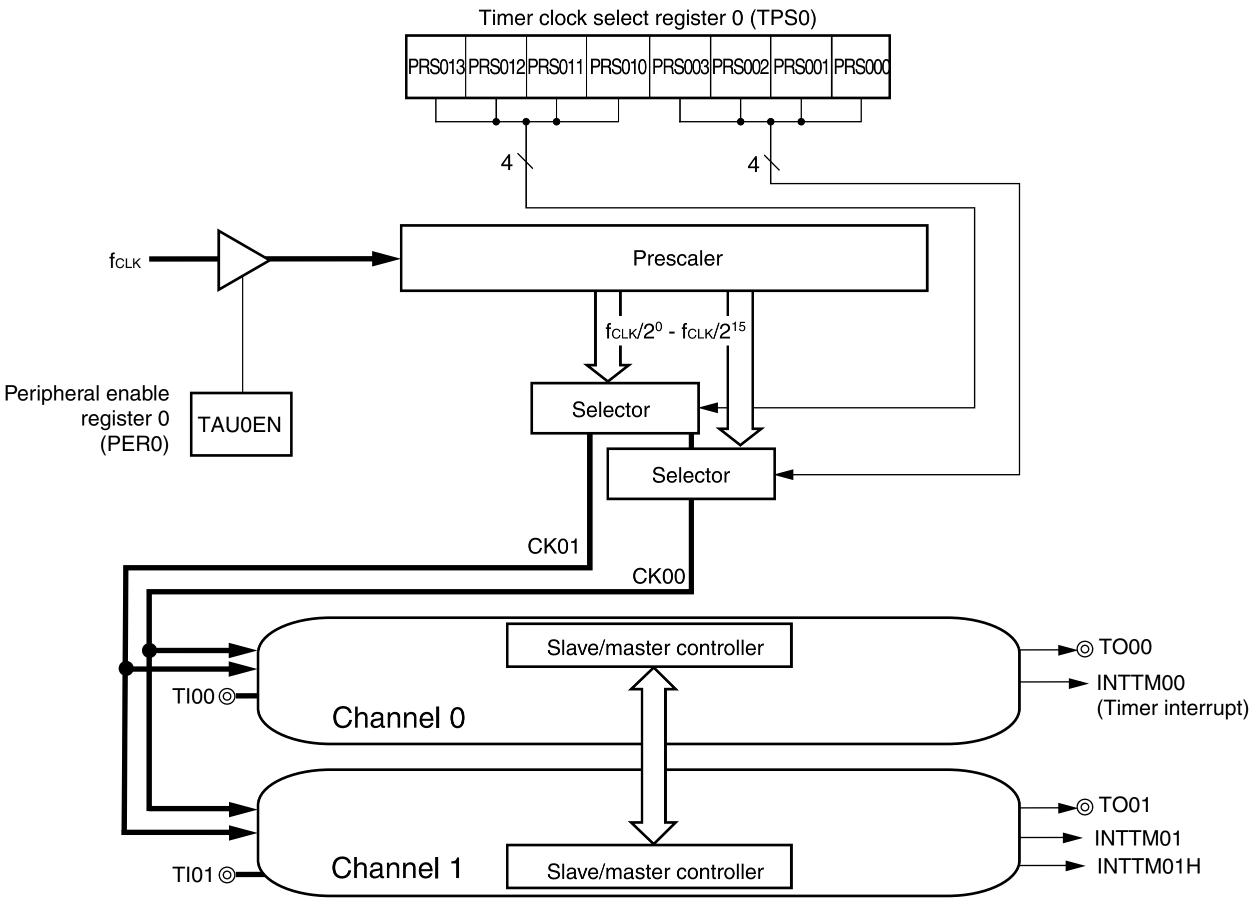
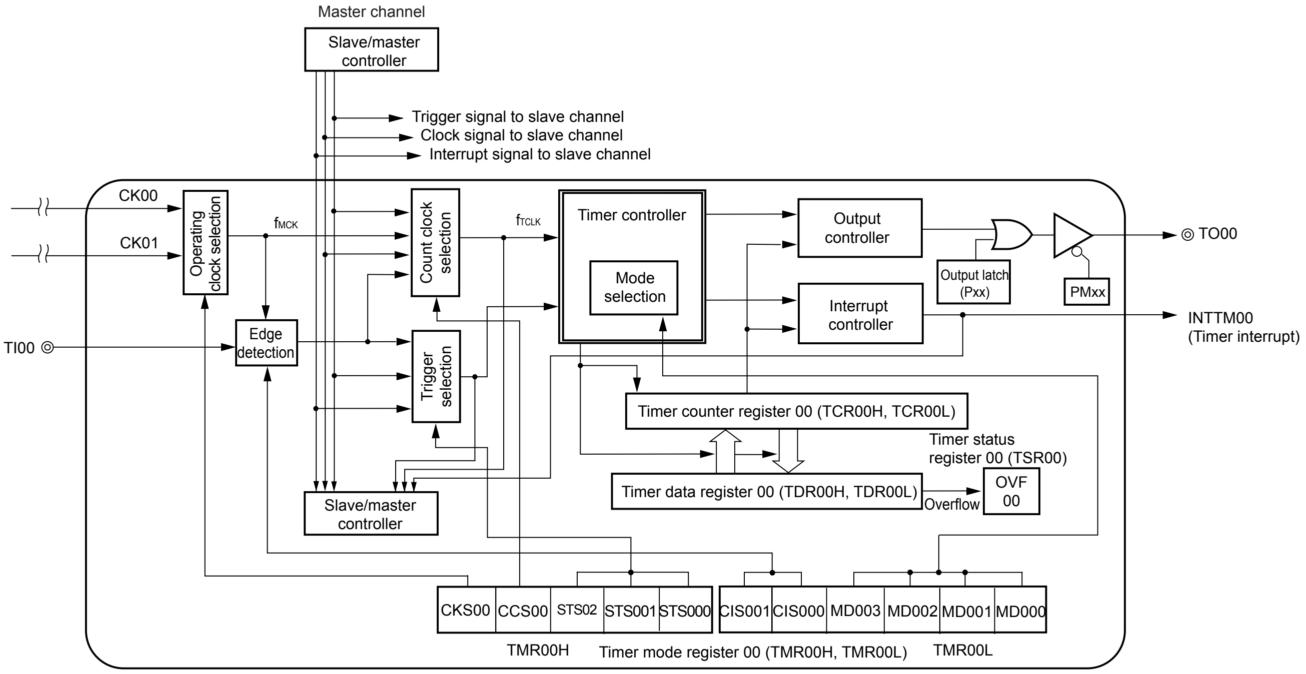
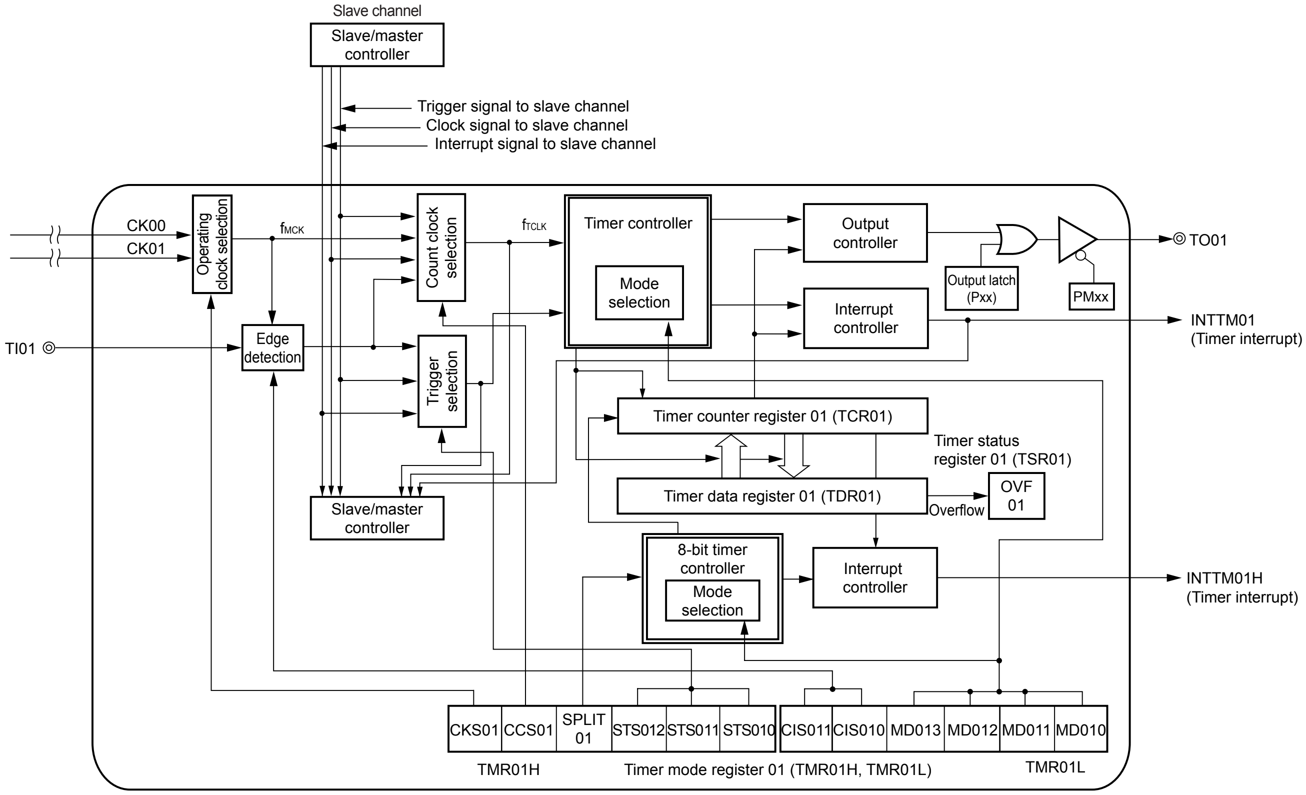


Figure 6-2. Internal Block Diagram of Channel of Timer Array Unit

(a) Channel 0



(b) Channel 1



6.2.1 Timer/counter register 0n (TCR0n)

TCR0n register consists of 8-bit read-only registers (TCR0nH and TCR0nL) and is used to count clocks.

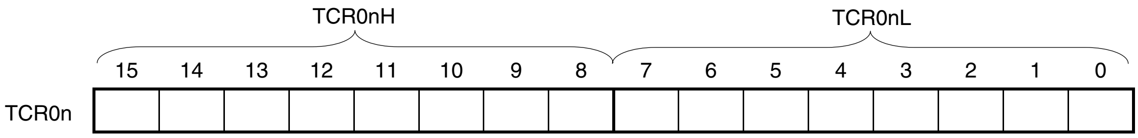
When data is read from the TCR0n register, the TCR0nH and TCR0nL registers must be accessed consecutively.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock.

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MD0n3 to MD0n0 bits of timer mode register 0n (TMR0n) (refer to **6.3.3 Timer mode register 0n (TMR0nH, TMR0nL) (n = 0 to 3)**).

Figure 6-3. Format of Timer/Counter Register 0n (TCR0n) (n = 0, 1)

Address: F0180H (TCR00L), F0181H (TCR00H) After reset: FFH R
: F0182H (TCR01L), F0183H (TCR01H)



Remark n: Channel number (n = 0, 1)

Reading from the TCR0nH and TCR0nL registers must be performed in a row with data in order of that from the TCR0nL register and that from the TCR0nH register.

If data are read from TCR0nL between the successive read, reading is not performed correctly.

Caution Consecutive reading from the TCR0nH and TCR0nL registers must be performed in the state where an interrupt is disabled by the DI instruction.

The count value can be read by reading timer counter register 0n (TCR0n).

The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAU0EN bit of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting 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

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

Cautions 1. The count value is not captured to timer data register 0n (TDR0n) even when the TCR0n register is read.

<R> **2. When channel 1 is used in 8-bit timer mode (SPLIT = 1), it is prohibited to read the TCR01H and TDR01H registers.**

The TCR0n register read value differs as follows according to operation mode changes and the operating status.

Table 6-2. Timer/counter Register 0n (TCR0n) Read Value in Various Operation Modes

Operation Mode	Count Mode	Timer/Counter Register 0n (TCR0n) Read Value[Note]			
		Value if the operation mode was changed after releasing reset	Value if the count operation paused (TT0n = 1)	Value if the operation mode was changed after count operation paused (TT0n = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Count down	FFFFH	Value if stop	Undefined	–
Capture mode	Count up	0000H	Value if stop	Undefined	–
Event counter mode	Count down	FFFFH	Value if stop	Undefined	–
One-count mode	Count down	FFFFH	Value if stop	Undefined	FFFFH
Capture & one-count mode	Count up	0000H	Value if stop	Undefined	Capture value of TDR0n register + 1

Note This indicates the value read from the TCR0n register when channel n has stopped operating as a timer (TE0n = 0) and has been enabled to operate as a counter (TS0n = 1). The read value is held in the TCR0n register until the count operation starts.

Remark n: Channel number (n = 0, 1)

Cautions 1. The count value is not captured to timer data register 0n (TDR0n) even when the TCR0n register is read.

<R> **2. When channel 1 is used in 8-bit timer mode (SPLIT = 1), it is prohibited to read the TCR01H and TDR01H registers.**

6.2.2 Timer data register 0n (TDR0n)

The TDR0 register consists of two eight bit registers (TCR0nH, TCR0nL) for which the capture or comparison functions can be selected.

Switching between the capture and comparison functions is by using the MD0n3 to MD0n0 bits of the timer mode register 0n (TMR0n) to select the operating mode.

<R> When using the TDR0n register as a compare register, the value of the TDR0nL and TDR0nH registers can be changed at any time.

For access to a TDR0n register, the TDR0nH and TDR0nL registers must be accessed consecutively.

In eight-bit timer mode (i.e. when the SPLIT0n bit of timer mode register 0n (TMR0n) is set to “1”), the TDR0n register can be rewritten in eight-bit units, with the higher 8 bits used as TDR0nH and the lower 8 bits used as TDR0nL.

The following points for caution apply when data are read from or written to TDR0nH and TDR0nL registers.

- In 16-bit timer mode (when channel 0 is in use, or bit 3 (SPLIT0n) of the TMR0nH register of channel 1 is cleared to “0”)
  Writing to TDR0nH and TDR0nL registers must be performed by writing in a row with data in order of that for the TDR0nH register and that for the TDR0nL register.
  Reading from the TDR0nH and TDR0nL registers must be performed in a row with data in order of that from the TDR0nL register and that from the TDR0nH register.
  If data are written to TDR0nH, read from TDR0nL, or read from TCR0n between the successive read or successive write operations, reading and writing is not performed correctly.
  <R> Consecutive reading from the TDR0nH and TDR0nL registers and consecutive writing to the TDR0nH and TDR0nL registers must be performed in the state where an interrupt is disabled by the DI instruction.

- In 8-bit timer mode (when bit 3 (SPLIT0n) of the TMR0nH register of channel 1 is set to “1”)
  The data can be written to the TDR0nH and TDR0nL registers in 8-bit units in 8-bit timer mode.
  Reading from TDR0nH register must be performed in a row with data in order of that from the TDR0nL register and that from the TDR0nH register.
  If data are written to TDR0nH, read from TDR0nL, or read from TCR0n between the successive read operations, reading is not performed correctly.
  Consecutive reading from the TDR0nH and TDR0nL registers must be performed in the state where an interrupt is disabled by the DI instruction.

Remark n: Channel number (n = 0, 1)

<R> **Caution When channel 1 is used in 8-bit timer mode (SPLIT = 1), it is prohibited to read the TCR01H and TDR01H registers.**

Figure 6-4. Format of Timer Data Register 0n (TDR0nH, TDR0nL) (n = 0)

Address: FFF18H (TDR00L), FFF19H (TDR00H), After reset: 00H R/W

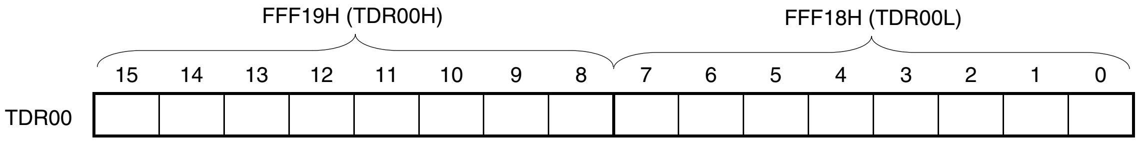
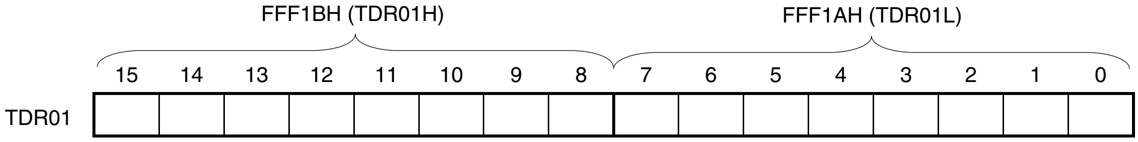


Figure 6-5. Format of Timer Data Register 0n (TDR0nH, TDR0nL) (n = 1)

Address: FFF1AH (TDR01L), FFF1BH (TDR01H), After reset: 00H R/W



(i) When timer data register 0n (TDR0nH, TDR0nL) is used as compare register

Counting down is started from the value set to the TDR0nH and TDR0nL registers. When the count value reaches 0000H, an interrupt signal (INTTM0n) is generated. The TDR0n register holds its value until it is rewritten.

Caution The TDR0n register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register 0n (TDR0nH, TDR0nL) is used as capture register

The count value of timer/counter register 0n (TCR0n) is captured to the TDR0nH and TDR0nL registers when the capture trigger is input.

A valid edge of the TI0n pin can be selected as the capture trigger. This selection is made by timer mode register 0n (TMR0n).

Remark n: Channel number (n = 0, 1)

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register 0 (TPS0)
- Timer channel enable status register 0 (TE0, TEH0)
- Timer channel start register 0 (TS0, TSH0)
- Timer channel stop register 0 (TT0, TTH0)
- Timer output enable register 0 (TOE0)
- Timer output register 0 (TO0)
- Timer output level register 0 (TOL0)
- Timer output mode register 0 (TOM0)
- Timer mode register 0n (TMR0nH, TMR0nL)
- Timer status register 0n (TSR0n)
- Noise filter enable register 1 (NFEN1)
- Port mode control register 0 (PMC0)
- Port mode registers 0, 4 (PM0, PM4)
- Port registers 0, 4 (P0, P4)

Remark n: Channel number (n = 0, 1)

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

Address: F00F0H After reset: 00H R/W

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

TAU0EN	Control of timer array 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 while the TAU0EN bit is set to 1 first. If TAU0EN = 0, writing to a control register of timer array unit is ignored, and all read values are default values (except for the noise filter enable register 1 (NFEN1), port mode registers 0, 4 (PM0, PM4), port registers 0, 4 (P0, P4), and port mode control register 0 (PMC0)).

- **Timer clock select register 0 (TPS0)**
- **Timer channel enable status register 0 (TE0, TEH0)**
- **Timer channel start register 0 (TS0, TSH0)**
- **Timer channel stop register 0 (TT0, TTH0)**
- **Timer output enable register 0 (TOE0)**
- **Timer output register 0 (TO0)**
- **Timer output level register 0 (TOL0)**
- **Timer output mode register 0 (TOM0)**
- **Timer mode register 0n (TMR0nH, TMR0nL)**
- **Timer status register 0n (TSR0n)**

2. Be sure to clear undefined bits to 0.

6.3.2 Timer clock select register 0 (TPS0)

The TPS0 register is a 8-bit register that is used to select two types of operation clocks (CK00, CK01) that are commonly supplied to each channel from external prescaler.

Rewriting of the TPS0 register during timer operation is possible only in the following cases.

If the PRS000 to PRS003 bits can be rewritten (n = 0 to 3):
  All channels for which CK00 is selected as the operation clock (CKS0n1 = 0) are stopped (TE0n = 0).
If the PRS010 to PRS013 bits can be rewritten (n = 0 to 3):
  All channels for which CK01 is selected as the operation clock (CKS0n1 = 1) are stopped (TE0n = 0).

The TPS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-7. Format of Timer Clock Select Register 0 (TPS0)

Address: F01B6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TPS0	PRS013	PRS012	PRS011	PRS010	PRS003	PRS002	PRS001	PRS000

PRS 0k3	PRS 0k2	PRS 0k1	PRS 0k0	Selection of operation clock (CK0k)^{Note} (k = 0, 1)					
					f_{CLK} = 1.25 MHz	f_{CLK} = 2.5 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	1.25 MHz	2.5 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	625 kHz	1.25 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	313 kHz	625 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	156 kHz	313 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	78.1 kHz	156 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	39.1 kHz	78.1 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	19.5 kHz	39.1 kHz	78.1 kHz	156 kHz	313 kHz
0	1	1	1	$f_{CLK}/2^7$	9.77 kHz	19.5 kHz	39.1 kHz	78.1 kHz	156 kHz
1	0	0	0	$f_{CLK}/2^8$	4.88 kHz	9.77 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	2.44 kHz	4.88 kHz	9.77 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.22 kHz	2.44 kHz	4.88 kHz	9.77 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	0	0	$f_{CLK}/2^{12}$	305 Hz	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	0	1	$f_{CLK}/2^{13}$	153 Hz	305 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	0	$f_{CLK}/2^{14}$	76.3 Hz	153 Hz	305 Hz	610 Hz	1.22 kHz
1	1	1	1	$f_{CLK}/2^{15}$	38.1 Hz	76.3 Hz	153 Hz	305 Hz	610 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TT0 = FFH).

Caution If f_{CLK} (undivided) is selected as the operation clock (CK0k) and TDR0m is set to 0000H (m = 0, 1), interrupt requests output from timer array units are not detected.

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

2. The above selected clock , but a signal which becomes high level for one period of f_{CLK} from its rising edge. For details, see **6.5.1 Count clock (f_{TCLK}).**

6.3.3 Timer mode register 0n (TMR0nH, TMR0nL)

The TMR0n register consists of two eight-bit registers (TMR0nH, TMR0nL) which set 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 channel 1), 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 TMR0nH and TMR0nL registers is prohibited when the register is in operation (when TE0n = 1).

The TMR0nH and TMR0nL registers can be set by a 8-bit memory manipulation instruction.

Reset signal generation clears TMR0nH and TMR0nL registers to 00H.

Caution The bits mounted depend on the channels in the bit 3 of TMR0nH register.
TMR01H: SPLIT01 bit
TMR01H: Fixed to 0

Figure 6-8. Format of Timer Mode Register 0n (TMR0n) (1/4)

Address: F0190H (TMR00L), F0191H (TMR00H) After reset: 00H R/W
: F0192H (TMR01L), F0193H (TMR01H)

Symbol	7	6	5	4	3	2	1	0
TMR01H	CKS011	0	0	CCS01	SPLIT01	STS012	STS011	STS010

Symbol	7	6	5	4	3	2	1	0
TMR00H	CKS001	0	0	CCS00	0	STS002	STS001	STS000

Symbol	7	6	5	4	3	2	1	0
TMR0nL (n = 0, 1)	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

CKS0n1	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CK00 set by timer clock select register 0 (TPS0)
1	Operation clock CK01 set by timer clock select register 0 (TPS0)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCS0n bit.	

CCS 0n	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKS0n0 and CKS0n1 bits
1	Valid edge of input signal input from the TI0n pin
Count clock (f_{TCLK}) is used for the timer/counter, output controller, and interrupt controller.	

Cautions 1. Be sure to clear the undefined bits to 0.

2. The timer array unit must be stopped (TT0 = 0FH) 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 CKS0n1 bit (f_{MCK}) or the valid edge of the signal input from the TI0n pin is selected as the count clock (f_{TCLK}).

Remark n: Channel number (n = 0, 1)

Figure 6-8. Format of Timer Mode Register 0n (TMR0n) (2/4)

Address: F0190H (TMR00L), F0191H (TMR00H) After reset: 00H R/W
: F0192H (TMR01L), F0193H (TMR01H)

Symbol	7	6	5	4	3	2	1	0
TMR01H	CKS011	0	0	CCS01	SPLIT01	STS012	STS011	STS010

Symbol	7	6	5	4	3	2	1	0
TMR00H	CKS001	0	0	CCS00	0	STS002	STS001	STS000

Symbol	7	6	5	4	3	2	1	0
TMR0nL (n = 0, 1)	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

(Bit 3 of TMR01H)

SPLIT01	Selection of 8 or 16-bit timer operation for channel 1
0	Operates as 16-bit timer.
1	Operates as 8-bit timer.

STS0n2	STS0n1	STS0n0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of the TI0n pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of the TI0n pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function).
Other than above			Setting prohibited

Note Bit 3 of TMR00H register is read-only and its value is fixed to 0. Writing is ignored.

Remark n: Channel number (n = 0, 1)

Figure 6-8. Format of Timer Mode Register 0n (TMR0n) (3/4)

Symbol	7	6	5	4	3	2	1	0
TMR0nL (n = 0, 1)	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

CIS 0n1	CIS 0n0	Selection of TI0n pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STS0n2 to STS0n0 bits is other than 010B, set the CIS0n1 to CIS0n0 bits to 10B.		

Remark n: Channel number (n = 0, 1)

Figure 6-8. Format of Timer Mode Register 0n (TMR0n) (4/4)

Symbol	7	6	5	4	3	2	1	0
TMR0nL (n = 0, 1)	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

MD 0n3	MD 0n2	MD 0n1	Setting of operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer / Square wave output / Divider function / PWM output (master)	Down count
0	1	0	Capture mode	Input pulse interval measurement	Up count
0	1	1	Event counter mode	External event counter	Down count
1	0	0	One-count mode	Delay counter / One-shot pulse output / PWM output (slave)	Down count
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Up count
Other than above			Setting prohibited		
The operation of MD0n0 bit changes depending on the operation of each mode (refer to the table bellow)					

Operation mode (Value set by the MD0n3 to MD0n1 bits (see table above))	MD 0n0	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 1] (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 2]. At that time, interrupt is not generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated.
Other than above		Setting prohibited

Notes **1.** In one-count mode, interrupt output (INTTM0n) when starting a count operation and TO0n output are not controlled.

2. If the start trigger (TS0n = 1) is issued during operation, the counter is initialized, and recounting is started (interrupt request is not generated).

Remark n: Channel number (n = 0, 1).

6.3.4 Timer status register 0n (TSR0n)

The TSR0n register indicates the overflow status of the counter of channel n.

The TSR0n register is valid only in the capture mode (MD0n3 to MD0n1 = 010B) and capture & one-count mode (MD0n3 to MD0n1 = 110B). It will not be set in any other mode. See Table 6-3 for the operation of the OVF bit in each operation mode and set/clear conditions.

The TSR0n register can be read by a 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-9. Format of Timer Status Register 0n (TSR0n)

Address: F01A0H (TSR00), F01A2H (TSR01) After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TMR0n	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.	

Table 6-3. 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	

Remarks1. The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

2. n: Channel number (n = 0, 1).

6.3.5 Timer channel enable status register 0 (TE0, TEH0 (8-bit mode))

The TE0 and TEH0 registers are used to enable or stop the timer operation of each channel.

Each bit of the TE0 and TEH0 registers correspond to each bit of the timer channel start register 0 (TS0, TSH0) and the timer channel stop register 0 (TT0, TTH0). When a bit of the TS0 and TSH0 registers is set to 1, the corresponding bit of TE0 and TEH0 is set to 1. When a bit of the TT0 and TTH0 registers is set to 1, the corresponding bit of TE0 and TEH0 is cleared to 0.

The TE0 and TEH0 registers can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears TE0 and TEH0 registers to 00H.

Figure 6-10. Format of Timer Channel Enable Status Register 0 (TE0)

Address: F01B0H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TE0	0	0	0	0	0	0	TE01	TE00

TE0n	Indication of operation enable/stop status of channel n (n = 0, 1)
0	Operation is stopped.
1	Operation is enabled.
Indicates operation enable/stop status of the 16-bit timer. TE01 bits indicate whether operation of the lower 8-bit timer is enabled or stopped when channels 1 is in the 8-bit timer mode.	

Figure 6-11. Format of Timer Channel Enable Status Register 0 (TEH0)

Address: F01B1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TEH0	0	0	0	0	0	0	TEH01	0

TEH 01	Indication of operation enable/stop status of channel 1
0	Operation is stopped.
1	Operation is enabled.
This bit indicates whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode	

Remark n: Channel number (n = 0, 1)

6.3.6 Timer channel start register 0 (TS0, TSH0 (8-bit mode))

The TS0 and TSH0 registers are trigger registers that are used to initialize timer/counter register 0n (TCR0n) and start the counting operation of each channel.

When a bit of these registers is set to 1, the corresponding bit of timer channel enable status register 0 (TE0, TEH0) is set to 1. The TSH0n and TS0n bits are immediately cleared when operation is enabled (TE0n = 1), because they are trigger bits.

The TS0 and TSH0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TS0 and TSH0 registers to 00H.

Figure 6-12. Format of Timer Channel Start Register 0 (TS0)

Address: F01B2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TS0	0	0	0	0	0	0	TS01	TS00

TS0n	Operation enable (start) trigger of channel n
0	No trigger operation
1	The TE0n bit is set to 1 and the count operation becomes enabled. The TCR0n register count operation start in the count operation enabled state varies depending on each operation mode (see **Table 6-4**). TS01 bit is the trigger to enable operation (start operation) of the lower 8-bit timer when channel 1is in the 8-bit timer mode.

Figure 6-13. Format of Timer Channel Start Register 0 (TSH0)

Address: F01B3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TSH0	0	0	0	0	0	0	TSH01	0

TSH 01	Operation enable (start) trigger of channel 1
0	No trigger operation
1	The TEH01 bit is set to 1 and the count operation becomes enabled. The TCR01 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-4**).
This bit is the trigger to enable operation (start operation) of the higher 8-bit timer when channel 1 is in the 8-bit timer mode	

Cautions 1. Be sure to clear undefined bits to 0.

2. When switching from a function that does not use TI0n pin input to one that does, the following wait period is required from when timer mode register 0n (TMR0n) is set until the TS0n bit is set to 1.
When the TI0n pin noise filter is enabled (TNFEN = 1): Four cycles of the operation clock (f_{MCK})
When the TI0n pin noise filter is disabled (TNFEN = 0): Two cycles of the operation clock (f_{MCK})

Remarks1. When the TS0 and TSH0 registers are read, 0 is always read.

2. n: Channel number (n = 0, 1).

6.3.7 Timer channel stop register 0 (TT0, TTH0 (8-bit mode))

The TT0 and TTH0 registers are trigger registers that are used to stop the counting operation of each channel.

When a bit of TT0 and TTH0 registers is set to 1, the corresponding bit of timer channel enable status register 0 (TE0, TEH0) is cleared to 0. The TT0n and TTH0n bits are immediately cleared when operation is stopped (TE0n, TTH0n = 0), because they are trigger bits.

The TT0 and TTH0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TT0 and TTH0 registers to 00H.

Figure 6-14. Format of Timer Channel Stop Register 0 (TT0)

Address: F01B4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TT0	0	0	0	0	0	0	TT01	TT00

TT0n	Operation stop trigger of channel n
0	No trigger operation
1	TE0n is cleared to 0. Operation is stopped (stop trigger is generated). TT01 bit is the trigger to stop operation of the lower 8-bit timer when channel 1 is in the 8-bit timer mode.

Figure 6-15. Format of Timer Channel Stop Register 0 (TTH0)

Address: F01B5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TTH0	0	0	0	0	0	0	TTH01	0

TTH 01	Operation stop trigger of channel 1
0	No trigger operation
1	TEH01 is cleared to 0. Operation is stopped (stop trigger is generated).
This bit is the trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode	

Caution Be sure to clear undefined bits to 0.

Remarks1 When the TT0 and TTH0 registers are read, 0 is always read.

2. n: Channel number (n = 0, 1).

6.3.8 Timer output enable register 0 (TOE0)

The TOE0 register is used to enable or disable timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TO0n bit of timer output register 0 (TO0) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TO0n).

The TOE0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-16. Format of Timer Output Enable Register 0 (TOE0)

Address: F01BAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TOE0n	0	0	0	0	0	0	TOE01	TOE00

TOE 0n	Timer output enable/disable of channel n
0	Disable output of timer. Without reflecting on TO0n bit timer operation, to fixed the output. Writing to the TO0n bit is enabled.
1	Enable output of timer. Reflected in the TO0n bit timer operation, to generate the output waveform. Writing to the TO0n bit is disabled (writing is ignored).

Caution Be sure to clear undefined bits to 0.

Remark n: Channel number (n = 0, 1).

6.3.9 Timer output register 0 (TO0)

The TO0 register is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TO0n) of each channel.

The TO0n bit oh this register can be rewritten by software only when timer output is disabled (TOE0n = 0). When timer output is enabled (TOE0n = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the TO0n alternate pin as a port function pin, set the corresponding TO0n bit to 0.

The TO0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-17. Format of Timer Output Register 0 (TO0)

Address: F01B8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	TO01	TO00

TO0n	Timer output of channel n
0	Timer output value is “0”.
1	Timer output value is “1”.

Caution **Be sure to clear undefined bits to 0.**

Remark n: Channel number (n = 0, 1).

6.3.10 Timer output level register 0 (TOL0)

The TOL0 register is a register that controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOE0n = 1) in the Slave channel output mode (TOM0n = 1). In the master channel output mode (TOM0n = 0), this register setting is invalid.

The TOL0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-18. Format of Timer Output Level Register 0 (TOL0)

Address: F01BCH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	0	0	TOL01	0

TOL 0n	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 undefined bits to 0.

Remarks 1. The timer output logic is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.

2. n: Channel number (n = 0, 1)

6.3.11 Timer output mode register 0 (TOM0)

The TOM0 register is used to control the timer output mode of each channel.

When a channel is used for the independent channel operation function, set the corresponding bit of the channel to be used to 0.

When a channel is used for the simultaneous channel operation function (PWM output or one-shot pulse output), set the corresponding bit of the master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOE0n = 1: n = 0, 1).

The TOM0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-19. Format of Timer Output Mode Register 0 (TOM0)

Address: F01BEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	TOM01	0

TOM 0n	Control of timer output mode of channel n
0	Used as the independent channel operation function
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTM00) of the master channel, and reset by the timer interrupt request signal (INTTM01) of the slave channel)

Caution Be sure to clear undefined bits to 0.

Remark n: Channel number (n = 0, 1).

6.3.12 Noise filter enable register 1 (NFEN1)

The NFEN1 register is used to set whether the noise filter can be used for the timer input signal to each channel.

Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal.

When the noise filter is ON, match detection and synchronization of the 4 clocks is performed with the CPU/peripheral hardware clock (f_{MCK}). When the noise filter is OFF, match detection and synchronization of the 2 clocks is performed with the CPU/peripheral hardware clock (f_{MCK}). For details, see **6.5.1 (2) When valid edge of input signal via the TI0n pin is selected (CCS0n = 1)** and **6.5.2 Start timing of counter**.

The NFEN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-20. Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	0	0	0	0	0	0	TNFEN01	TNFEN00

TNFEN0n	Enable/disable using noise filter of TI0n pin input signal
0	Noise filter OFF
1	Noise filter ON

Remark n: Channel number (n = 0, 1)

6.3.13 Port mode register 0 (PM0)

This register sets input/output of port 0 in 1-bit units.

When using the ports that share the pin with the timer output (such as P04/ANI3/TI01/TO01/KR5) for timer output, set the bit in the port mode register 0 (PM0), the port register 0 (P0), and the port mode control register 0 (PMC0) corresponding to each port to 0.

Example: When using P04/ANI3/TI01/TO01/KR5 for timer output
Set the PMC04 bit of port mode control register 0 to 0.
Set the PM4 bit of port mode register 0 to 0.
Set the P04 bit of port register 0 to 0.

When using the ports (such as P04/ANI3/TI01/TO01/KR5) to be shared with the timer output pin for timer input, set the bit in the port mode register 4 (PM4) corresponding to each port to 1. Also set the bit in the port mode control register 4 (PMC4) corresponding to each port to 0. At this time, the bit in the port register 4 (P4) may be 0 or 1.

The PM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Remark TI01, TO00, and TO01 pins alternate analog input pins. When using the timer I/O function, the corresponding bit of the PMCx register for switching digital I/O or analog input is sure to set to “0”.

Figure 6-21. Format of Port Mode Register 0 (PM0)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W

PM0m	P0m pin I/O mode selection (m = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note When peripheral I/O modules are changed by the peripheral I/O redirection register (PIOR), PM4 is used.

Remark n: Channel number (n = 0, 1)

6.4 Basic Rules of Timer Array Unit

6.4.1 Basic rules of simultaneous channel operation function

When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

(1) Only an even channel (channel 0) can be set as a master channel.
(2) Channel 1 can be set as a slave channel.
(3) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKS011 bit (bit 7 of timer mode register 01H (TMR01H)) of the slave channel that operate in combination with the master channel must be the same value as that of the master channel.
(4) A master channel can transmit INTTM00 (interrupt), start software trigger, and count clock to the lower channels.
(5) A slave channel can use INTTM00 (interrupt), a start software trigger, or the count clock of the master channel as a source clock.
(6) To simultaneously start channels that operate in combination, the channel start trigger bit (TS0n) of the channels in combination must be set at the same time.
(7) During the counting operation, a TS0n bit of a master channel or TS0n bits of all channels which are operating simultaneously can be set. It cannot be applied to TS0n bits of slave channels alone.
(8) To stop the channels in combination simultaneously, the channel stop trigger bit (TT0n) of the channels in combination must be set at the same time.
(9) Timer mode register 0n (TMR0nH) 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.

Regarding point (2), the eight lower bits of channel 1 are selectable as the slave channel. In this case, the eight higher bits of channel 1 can be used as an interval timer.

Remark n: Channel number (n = 0, 1).

6.4.2 Basic rules of 8-bit timer operation function (only channel 1)

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 channel 1, 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 channel 1.
(2) When using 8-bit timers, set the SPLIT bit of timer mode register 01 (TMR01H) to 1.
(3) The higher 8 bits can be operated as the interval timer function.
(4) At the start of operation, the higher 8 bits output INTTM01H (an interrupt) (which is the same operation performed when MD010 is set to 1).
(5) The operation clock of the higher 8 bits is selected according to the CKS011 bit of the lower-bit TMR01H register.
(6) For the higher 8 bits, the TSH01 bit is manipulated to start channel operation and the TTH01 bit is manipulated to stop channel operation. The channel status can be checked using the TEH01 bit.
(7) The lower 8 bits operate according to the settings of TMR01H and TMR01L registers. The following four functions support operation of the lower 8 bits:
    - Interval timer function
    - External event counter function
    - Delay count function
    - PWM output

(8) For the lower 8 bits, the TS01 bit is manipulated to start channel operation and the TT01 bit is manipulated to stop channel operation. The channel status can be checked using the TE01 bit.
(9) During 16-bit operation, manipulating the TSH01/TTH01 bits is invalid. The TS01 and TT01 bits are manipulated to operate channel n. The TEH01 bit is not changed.
(10) For the 8-bit timer function, the simultaneous operation functions (one-shot pulse) cannot be used.

Remark n: Channel number (n = 0, 1).

<R> **Caution When channel 1 is used in 8-bit timer mode (SPLIT = 1), it is prohibited to read the TCR01H and TDR01H registers.**

6.5 Operation of Counter

6.5.1 Count clock (f_{TCLK})

The count clock (f_{TCLK}) of the timer array unit can be selected between following by CCS0n bit of timer mode register 0n (TMR0n). .

- Operation clock (f_{MCK}) specified by the CKS0n1 bit
- Valid edge of input signal input from the TI0n pin

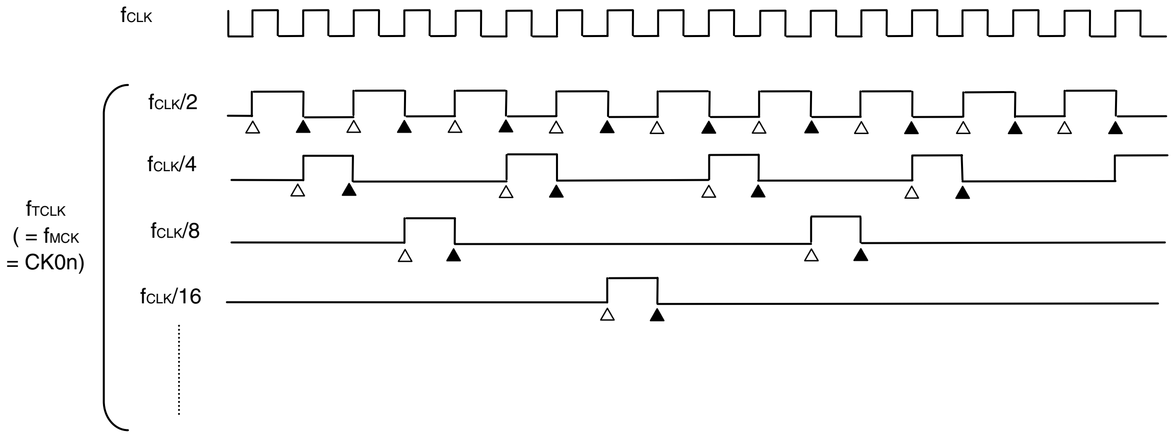
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

(1) When operation clock (f_{MCK}) specified by the CKS0n1 bit is selected (CCS0n = 0)

The count clock (f_{TCLK}) is between f_{CLK} to $f_{CLK}/2^{15}$ by setting of timer clock select register 0 (TPS0). When a divided f_{CLK} is selected, however, the clock selected in TPS0 register is at the high level for one period of f_{CLK} from its rising edge. When a f_{CLK} is selected, it is fixed to the high level

Counting of timer count register 0n (TCR0n) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as “counting at rising edge of the count clock”, as a matter of convenience.

Figure 6-22. Timing of f_{CLK} and Count Clock (f_{TCLK}) (When CCS0n = 0)



Remarks 1. △ : Rising edge of the count clock

▲ : Synchronization, increment/decrement of counter

2. f_{CLK}: CPU/peripheral hardware clock

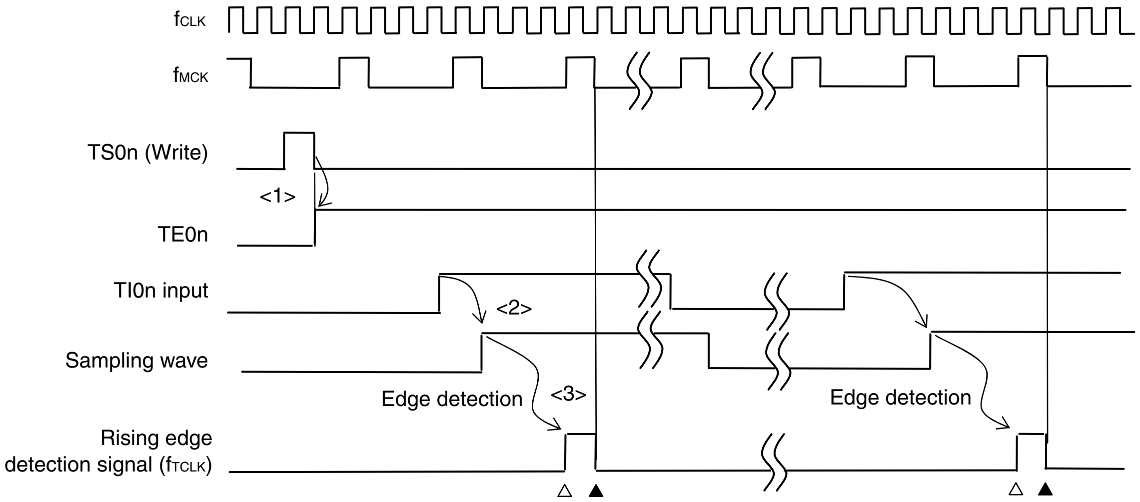
3 n: Channel number (n = 0, 1).

(2) When valid edge of input signal via the TI0n pin is selected (CCS0n = 1)

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TI0n pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 period of f_{MCK} from the input signal via the TI0n pin (when a noise filter is used, the delay becomes 3 to 4 clock).

Counting of timer count register 0n (TCR0n) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as "counting at valid edge of input signal via the TI0n pin", as a matter of convenience.

Figure 6-23. Timing of f_{CLK} and Count Clock (f_{TCLK}) (When CCS0n = 1, noise filter unused)



<1> Setting TS0n bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TI0n pin.

<2> The rise of input signal via the TI0n pin is sampled by f_{MCK}.

<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Remarks 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter

2. f_{CLK}: CPU/peripheral hardware clock
f_{MCK}: Operation clock of channel n

3. The waveform of the input signal via TI0n pin of the input pulse interval measurement, the measurement of high/low width of input signal, and the delay counter, the one-shot pulse output are the same as that shown in **Figure 6-23**.

4 n: Channel number (n = 0, 1).

6.5.2 Start timing of counter

Timer count register 0n (TCR0n) becomes enabled to operation by setting of TS0n bit of timer channel start register 0 (TS0).

Operations from count operation enabled state to timer count Register 0n (TCR0n) count start is shown in **Table 6-4**.

Table 6-4. Operations from Count Operation Enabled State to Timer count Register 0n (TCR0n) Count Start

Timer Operation Mode	Operation When TS0n = 1 Is Set
• Interval timer mode	No operation is carried out from start trigger detection (TS0n = 1) until count clock generation. The first count clock loads the value of the TDR0n register to the TCR0n register and the subsequent count clock performs count down operation (see **6.5.3 (1) Interval timer mode operation**).
• Event counter mode	Writing 1 to the TS0n bit loads the value of the TDR0n register to the TCR0n register. Detection TI0n input edge, the subsequent count clock performs count down operation. (see **6.5.3 (2) Event counter mode operation**).
• Capture mode	No operation is carried out from start trigger (TS0n = 1) detection until count clock generation. The first count clock loads 0000H to the TCR0n register and the subsequent count clock performs count up operation (see **6.5.3 (3) Capture mode operation (input pulse interval measurement)**).
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TS0n bit while the timer is stopped (TE0n = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDR0n register to the TCR0n register and the subsequent count clock performs count down operation (see **6.5.3 (4) One-count mode operation**).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TS0n bit while the timer is stopped (TE0n = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCR0n register and the subsequent count clock performs count up operation (see **6.5.3 (5) Capture & one-count mode operation (high-level width is measured)**).

Remark n: Channel number (n = 0, 1).

6.5.3 Counter Operation

Here, the counter operation in each mode is explained.

(1) Interval timer mode operation

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit. Timer count register 0n (TCR0n) holds the initial value until count clock generation.

<2> A start trigger is generated at the first count clock after operation is enabled.

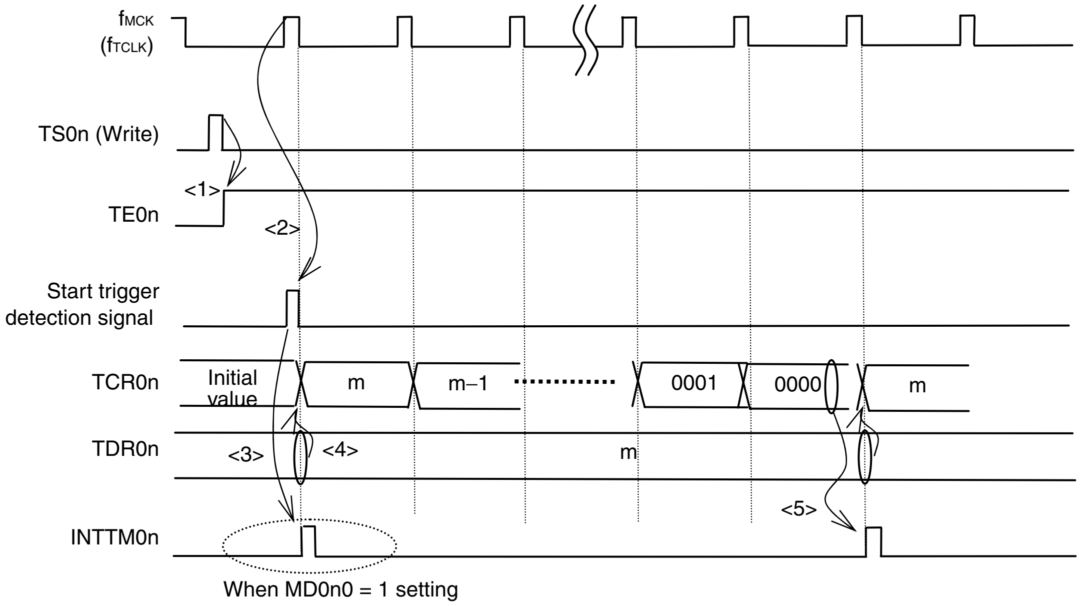
<3> When the MD0n0 bit is set to 1, INTTM0n is generated by the start trigger.

<4> By the first count clock after the operation enable, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting starts in the interval timer mode.

<5> When the TCR0n register counts down and its count value is 0000H, INTTM0n is generated in the next count clock (f_{MCK}) and the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting keeps on.

Figure 6-24. Operation Timing (In Interval Timer Mode)

<R>



Caution In the first cycle operation of count clock after writing the TS0n bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MD0n0 = 1.

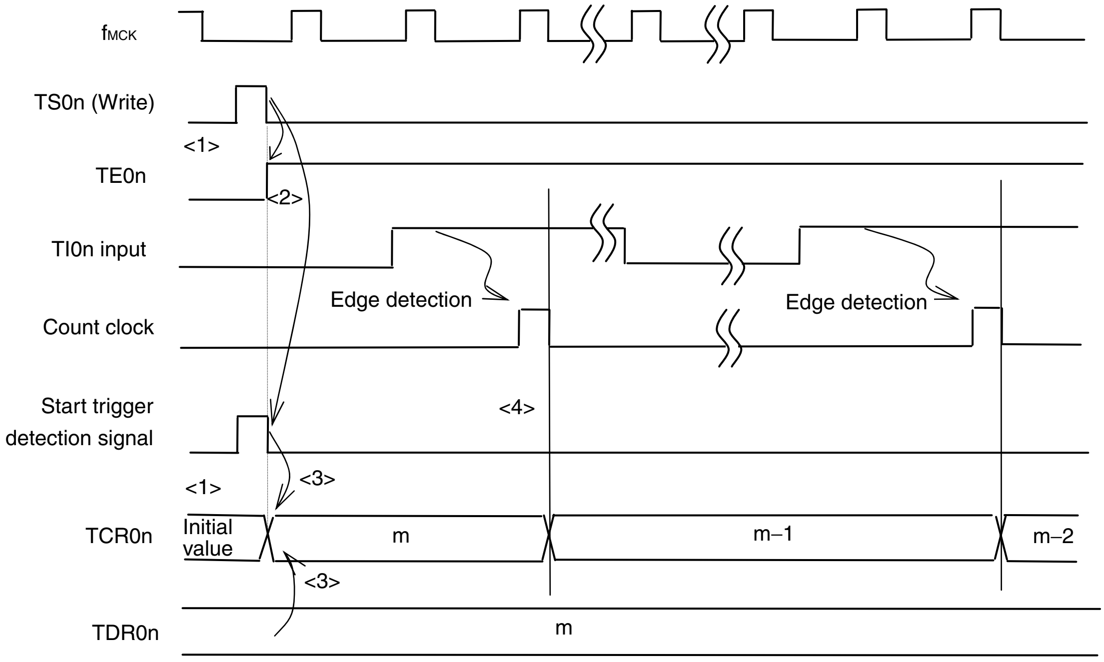
Remarks1. f_{MCK}, the start trigger detection signal, and INTTM0n become active between one clock in synchronization with f_{CLK}.

2. n: Channel number (n = 0, 1).

(2) Event counter mode operation

<1> Timer count register 0n (TCR0n) holds its initial value while operation is stopped (TE0n = 0).
<2> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.
<3> As soon as 1 has been written to the TS0n bit and 1 has been set to the TE0n bit, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register to start counting.
<4> After that, the TCR0n register value is counted down according to the count clock of the valid edge of the TI0n input.

Figure 6-25. Operation Timing (In Event Counter Mode)



Remarks1. The timing is shown in **Figure 6-25** 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 TI0n input.
The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (f_{MCK}).

2. n: Channel number (n = 0, 1).

(3) Capture mode operation (input pulse interval measurement)

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.

<2> Timer count register 0n (TCR0n) holds the initial value until count clock generation.

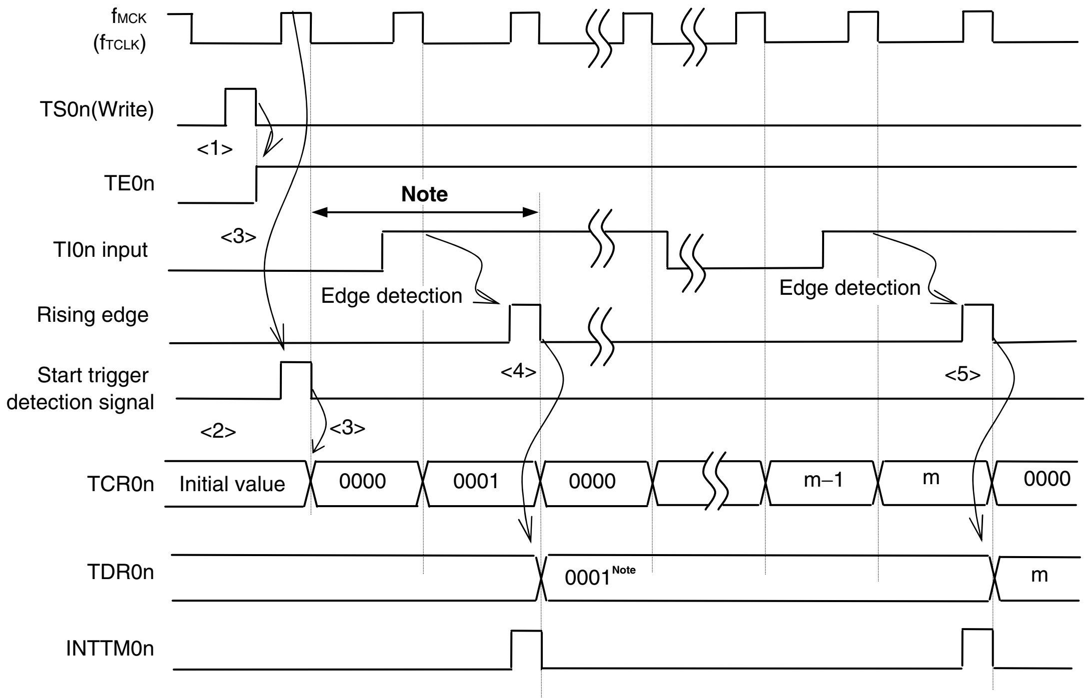
<3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0000H is loaded to the TCR0n register and counting starts in the capture mode. (When the MD0n0 bit is set to 1, INTTM0n is generated by the start trigger.)

<4> On detection of the valid edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and INTTM0n is generated. However, this capture value has no meaning. The TCR0n register keeps on counting from 0000H.

<5> On next detection of the valid edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and INTTM0n is generated.

Figure 6-26. Operation Timing (In Capture Mode: Input Pulse Interval Measurement)

<R>



Caution In the first cycle operation of count clock after writing the TS0n bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MD0n0 = 1.

Note If a clock has been input to TI0n (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

Remarks 1. The timing is shown in **Figure 6-26** 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 TI0n input.
The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (fMCK).

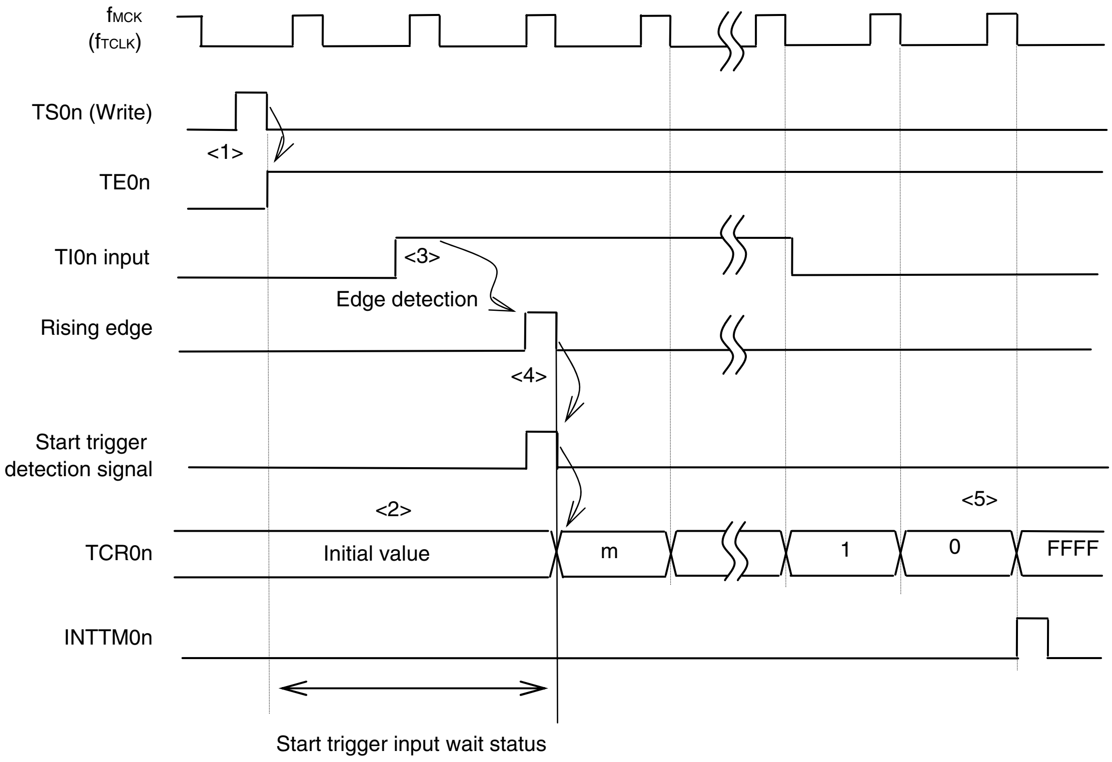
2. n: Channel number (n = 0, 1).

(4) One-count mode operation

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.
<2> Timer count register 0n (TCR0n) holds the initial value until start trigger generation.
<3> Rising edge of the TI0n input is detected.
<4> On start trigger detection, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and count starts.
<5> When the TCR0n register counts down and its count value is 0000H, INTTM0n is generated and the value of the TCR0n register becomes FFFFH and counting stops.

Figure 6-27. Operation Timing (In One-count Mode)

<R>



Remarks1. 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 TI0n input. The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (f_{MCK}).

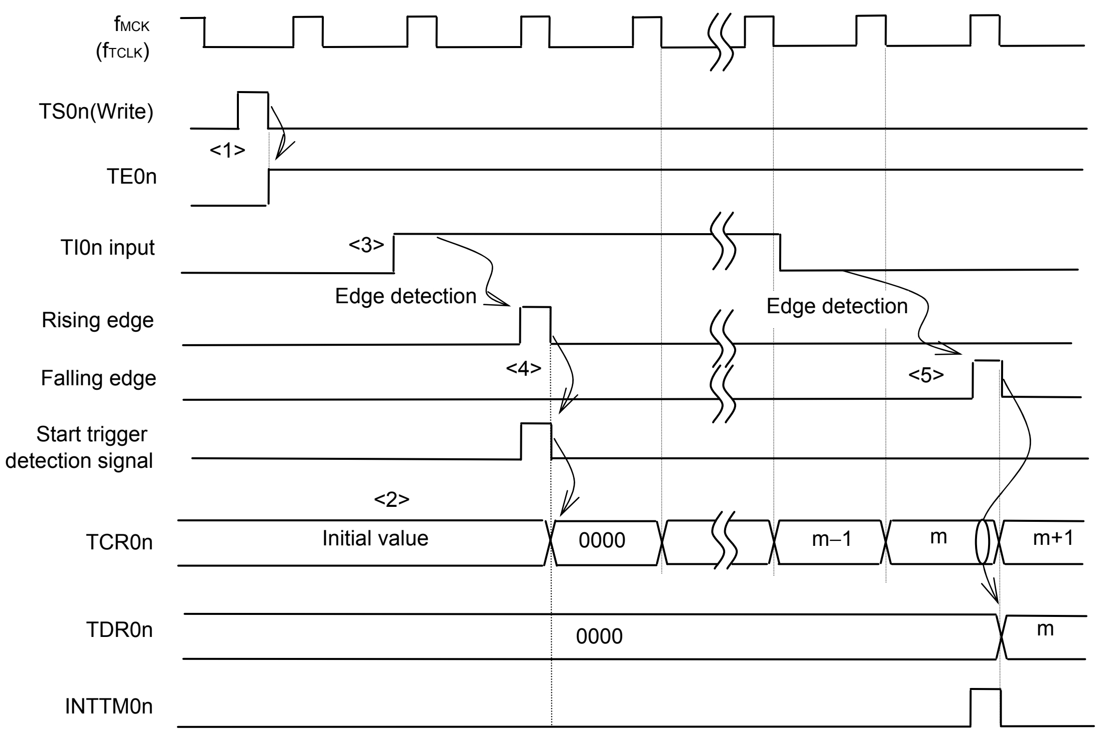
2. n: Channel number (n = 0, 1).

(5) Capture & one-count mode operation (high-level width is measured)

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit of timer channel start register 0 (TS0).
<2> Timer count register 0n (TCR0n) holds the initial value until start trigger generation.
<3> Rising edge of the TI0n input is detected.
<4> On start trigger detection, the value of 0000H is loaded to the TCR0n register and count starts.
<5> On detection of the falling edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and INTTM0n is generated.

Figure 6-28. Operation Timing (In Capture & One-count Mode: High-level Width Measurement)

<R>



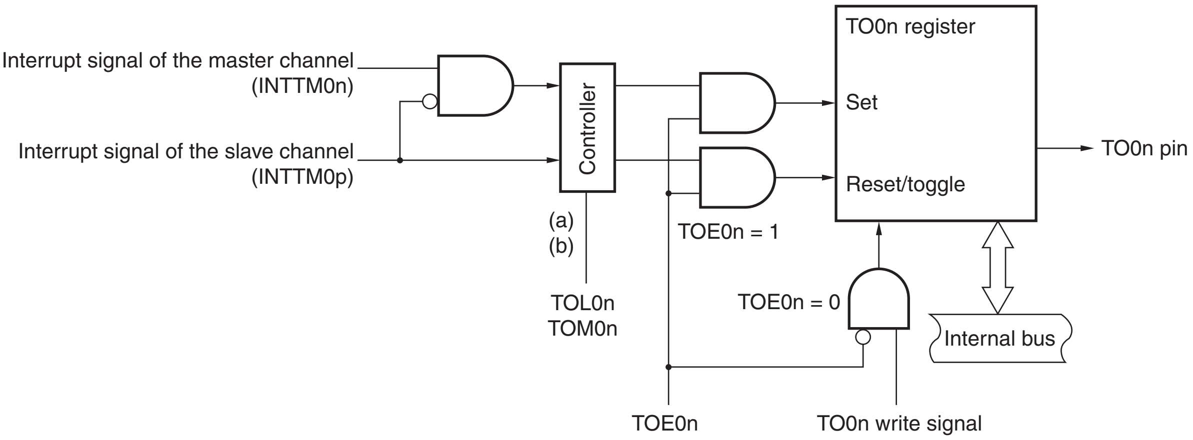
Remark1. The timing is shown in **Figure 6-28** indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TI0n input. The error per one period occurs be the asynchronous between the period of the TI0n input and that of the count clock (fMCK).

2. n: Channel number (n = 0, 1).

6.6 Channel Output (TO0n pin) Control

6.6.1 TO0n pin output circuit configuration

Figure 6-29. Output Circuit Configuration



The following describes the TO0n pin output circuit.

<1> While timer output is enabled (TOE0n = 1), INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0 register. Writing to the TO0 register (TO0n write signal) becomes invalid.

When TOE0n = 1, the TO0n pin output never changes with signals other than interrupt signals.

To initialize the TO0n pin output level, it is necessary to set timer operation is stopped (TOE0n = 0) and to write a value to the TO0 register.

(a) When TOM0n = 0 (master channel output mode), the set value of timer output level register 0 (TOL0) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register 0 (TO0).

(b) When TOM0n = 1 (slave channel output mode), both INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0 register.

At this time, the TOL0 register becomes valid and the signals are controlled as follows:

When TOL0n = 0: Positive logic output (INTTM0n → set, INTTM0p → reset)
When TOL0n = 1: Negative logic output (INTTM0n → reset, INTTM0p → set)

When INTTM0n and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTM0n (set signal) is masked.

<2> While timer output is disabled (TOE0n = 0), writing to the TO0n bit to the target channel (TO0n write signal) becomes valid. When timer output is disabled (TOE0n = 0), neither INTTM0n (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TO0 register.

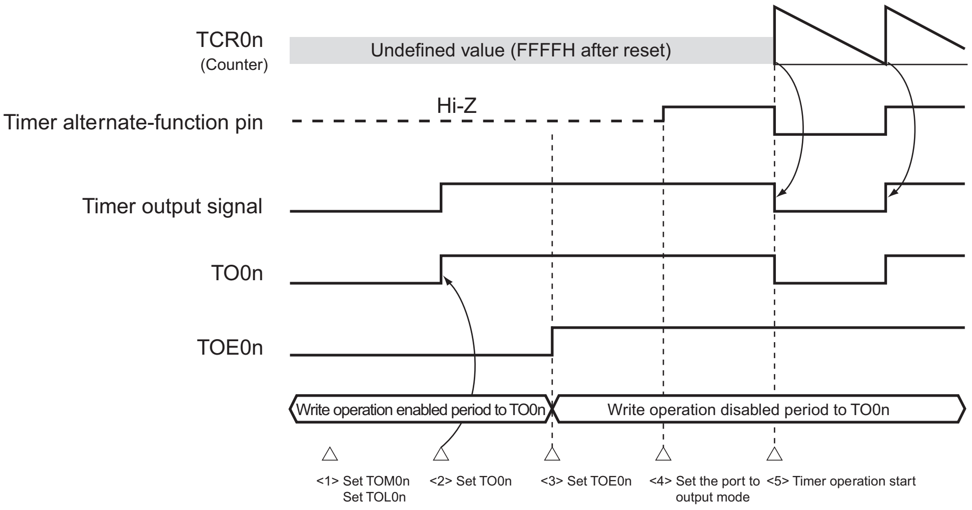
<3> The TO0 register can always be read, and the TO0n pin output level can be checked.

Remark n: Channel number n = 0, 1 (n = 0 for master channel)
p: Slave channel number p = 1

6.6.2 TO0n pin output setting

The following figure shows the procedure and status transition of the TO0n output pin from initial setting to timer operation start.

Figure 6-30. Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOM0n bit (0: Master channel output mode, 1: Slave channel output mode)
- TOL0n bit (0: Positive logic output, 1: Negative logic output)

<2> The timer output signal is set to the initial status by setting timer output register 0 (TO0).

<3> The timer output operation is enabled by writing 1 to the TOE0n bit (writing to the TO0 register is disabled).

<4> The port is set to digital I/O by port mode control register (PMCxx) (see **6.3.13 Port mode register 0 (PM0)**).

<5> The port I/O setting is set to output (see **6.3.13 Port mode register 0 (PM0)**).

<6> The timer operation is enabled (TS0n = 1).

Remark n: Channel number (n = 0, 1)

6.6.3 Cautions on channel output operation

(1) Changing values set in the registers TO0, TOE0, and TOL0 during timer operation

Since the timer operations (operations of timer count register 0n (TCR0n) and timer data register 0n (TDR0n)) are independent of the TO0n output circuit and changing the values set in timer output register 0 (TO0), timer output enable register 0 (TOE0), timer output level register 0 (TOL0) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TO0n pin by timer operation, however, set the TO0, TOE0, TOL0, and TOM0 registers to the values stated in the register setting example of each operation.

When the values set to the TOE0 and TOL0 registers (but not the TO0 register) are changed close to the occurrence of the timer interrupt (INTTM0n) of each channel, the waveform output to the TO0n pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTM0n) occurs.

Remark n: Channel number (n = 0, 1)

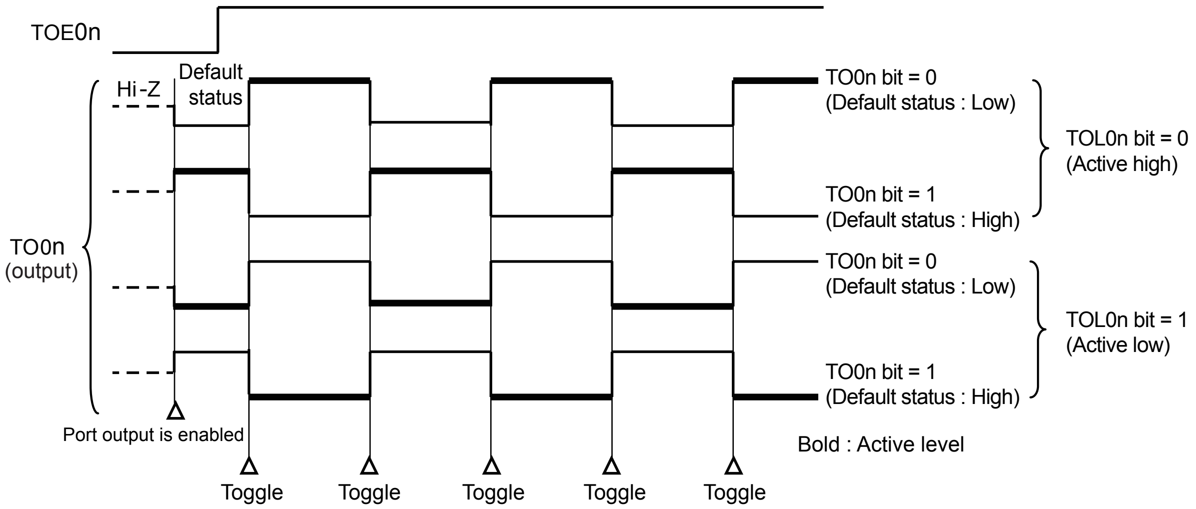
(2) Default level of TO0n pin and output level after timer operation start

The change in the output level of the TO0n pin when timer output register 0 (TO0) is written while timer output is disabled (TOE0n = 0), the initial level is changed, and then timer output is enabled (TOE0n = 1) before port output is enabled, is shown below.

(a) When operation starts with master channel output mode (TOM0n = 0) setting

The setting of timer output level register 0 (TOL0) is invalid when master channel output mode (TOM0n = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TO0n pin is reversed.

Figure 6-31. TO0n Pin Output Status at Toggle Output (TOM0n = 0)



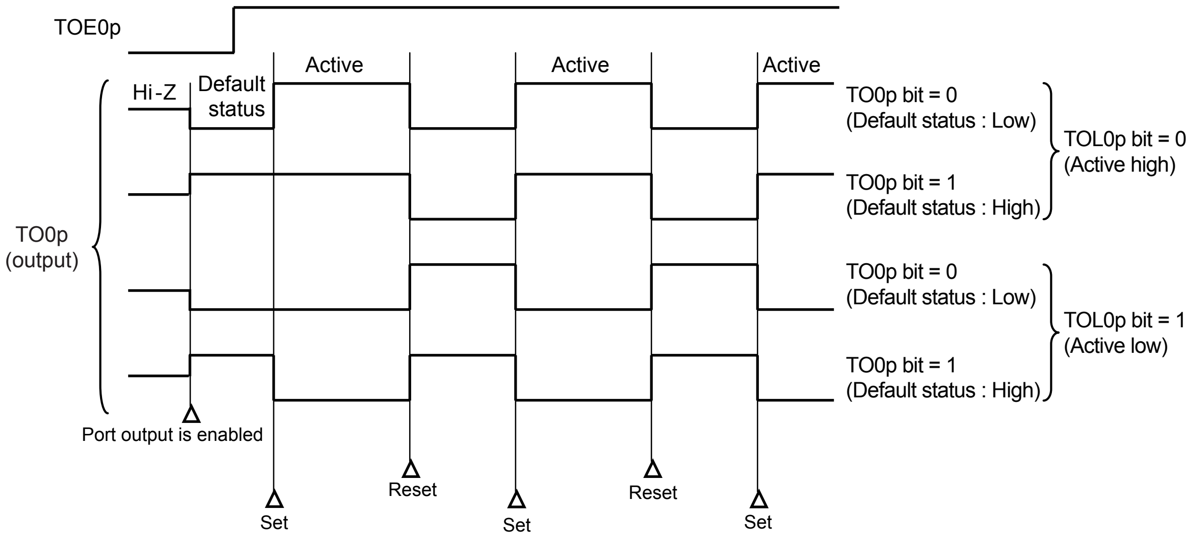
Remarks 1. Toggle: Reverse TO0n pin output status

2. n: Channel number (n = 0, 1).

(b) When operation starts with slave channel output mode (TOM0p = 1) setting (PWM output))

When slave channel output mode (TOM0p = 1), the active level is determined by timer output level register 0 (TOL0p) setting.

Figure 6-32. TO0p Pin Output Status at PWM Output (TOM0p = 1)



Remarks 1. Set: The output signal of the TO0p pin changes from inactive level to active level.
Reset: The output signal of the TO0p pin changes from active level to inactive level.

2. p: Channel number (p = 1)

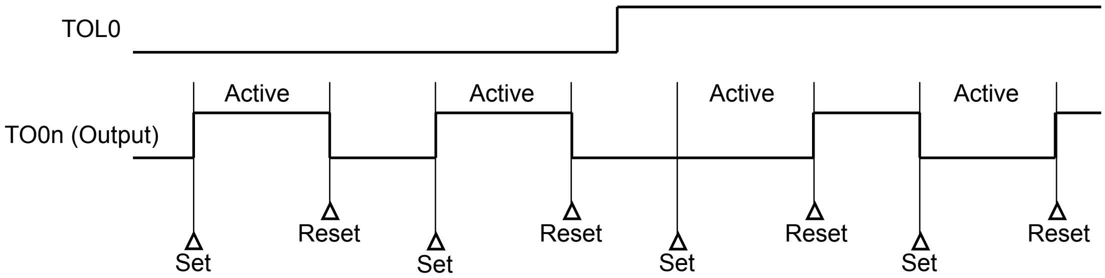
(3) Operation of TO0n pin in slave channel output mode (TOM0n = 1)

(a) When timer output level register 0 (TOL0) setting has been changed during timer operation

When the TOL0 register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TO0n pin change condition. Rewriting the TOL0 register does not change the output level of the TO0n pin.

The operation when TOM0n is set to 1 and the value of the TOL0 register is changed while the timer is operating (TE0n = 1) is shown below.

Figure 6-33. Operation when TOL0 Register Has Been Changed during Timer Operation



Remarks 1. Set: The output signal of the TO0n pin changes from inactive level to active level.
Reset: The output signal of the TO0n pin changes from active level to inactive level.

2. n: Channel number (n = 0, 1).

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TO0n pin/TO0n bit set timing at master channel timer interrupt (INTTM0n) generation is delayed by 1 count clock by the slave channel.

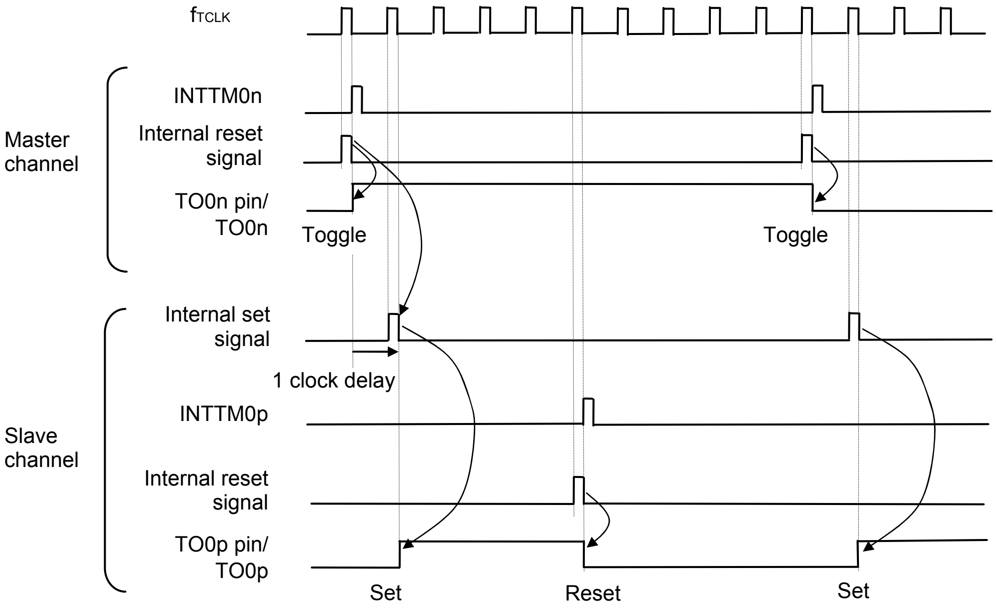
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-34 shows the set/reset operating statuses where the master/slave channels are set as follows.

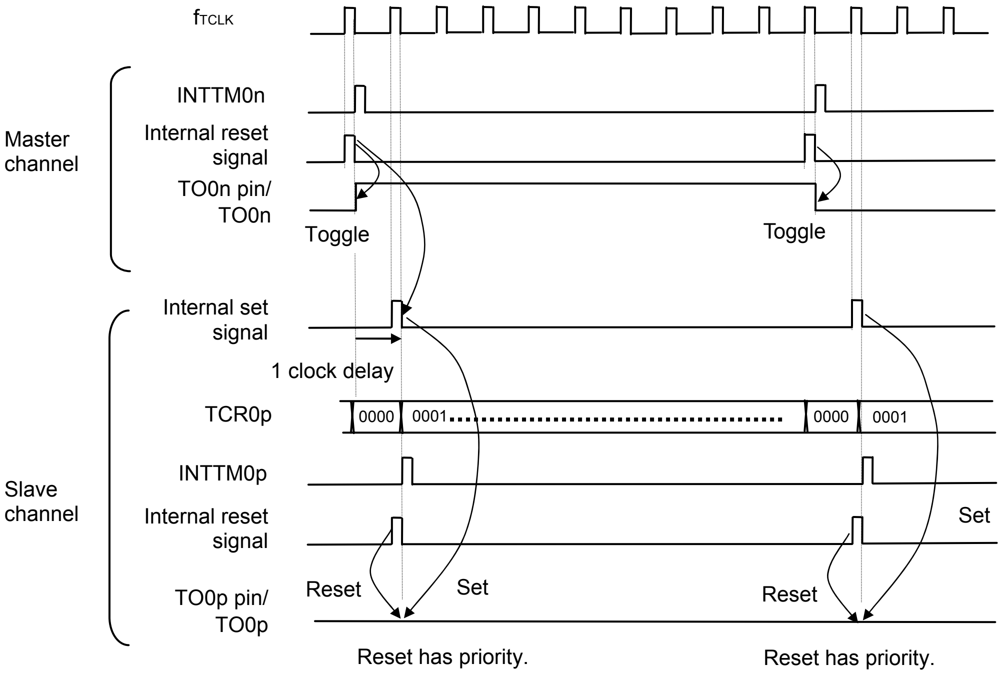
Master channel:	TOE0n = 1, TOM0n = 0, TOL0n = 0
Slave channel:	TOE0p = 1, TOM0p = 1, TOL0p = 0

Figure 6-34. Set/Reset Timing Operating Statuses

(a) Basic operation timing



(b) Operation timing when 0 % duty



Remarks 1. Internal reset signal: TO0n pin reset/toggle signal

Internal set signal: TO0n pin set signal

2. n: Master channel number (n = 0)

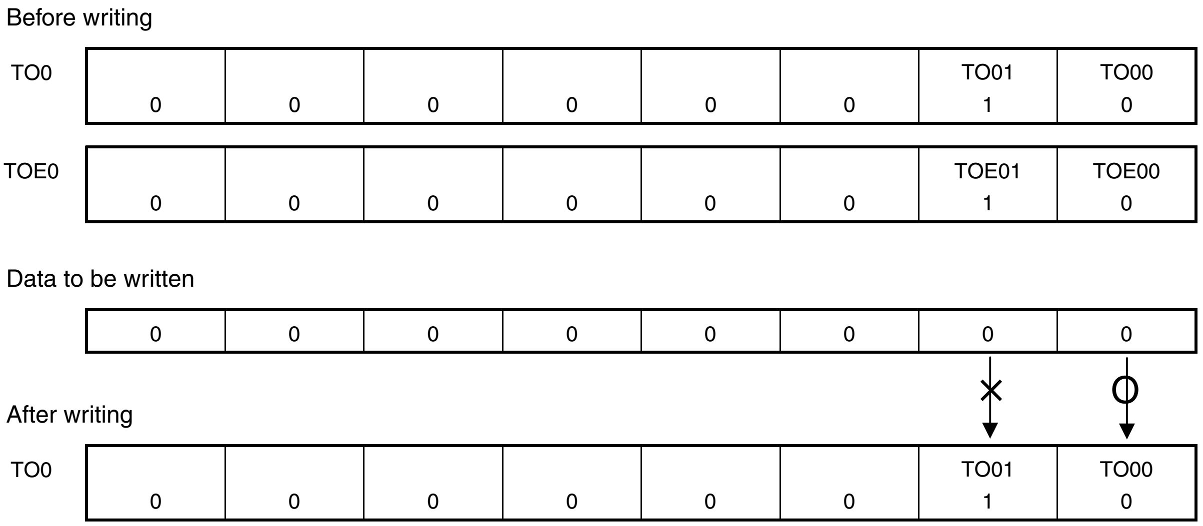
p: Slave channel number (p = 1)

6.6.4 Collective manipulation of TO0n bit

In timer output register 0 (TO0), the setting bits for all the channels are located in one register in the same way as timer channel start register 0 (TS0). Therefore, the TO0n bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TO0n bits (TOE0n = 0) that correspond to the relevant bits of the channel used to perfor0 output (TO0n).

Figure 6-35. Example of TO0n Bit Collective Manipulation



Before writing

TO0	0	0	0	0	0	0	TO01 1	TO00 0
TOE0	0	0	0	0	0	0	TOE01 1	TOE00 0

Data to be written

0	0	0	0	0	0	0	0
						×	O

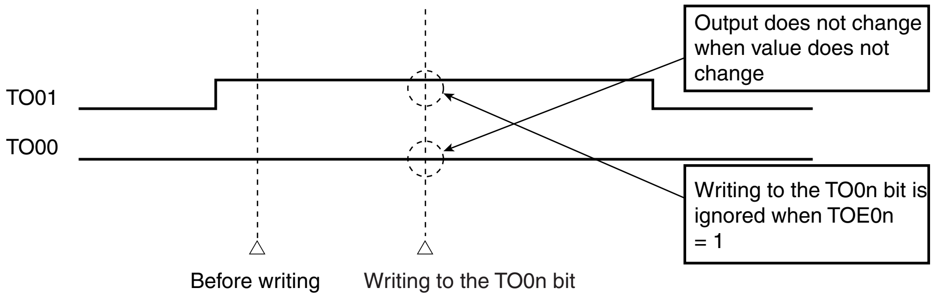
After writing

TO0	0	0	0	0	0	0	TO01 1	TO00 0

Writing is done only to the TO0n bit with TOE0n = 0, and writing to the TO0n bit with TOE0n = 1 is ignored.

TO0n (channel output) to which TOE0n = 1 is set is not affected by the write operation. Even if the write operation is done to the TO0n bit, it is ignored and the output change by timer operation is normally done.

Figure 6-36. TO0n Pin Statuses by Collective Manipulation of TO0n Bit



Remark n: Channel number (n = 0, 1).

6.6.5 Timer interrupt and TO0n pin output at operation start

In the interval timer mode or capture mode, the MD0n0 bit in timer mode register 0n (TMR0n) sets whether or not to generate a timer interrupt at count start.

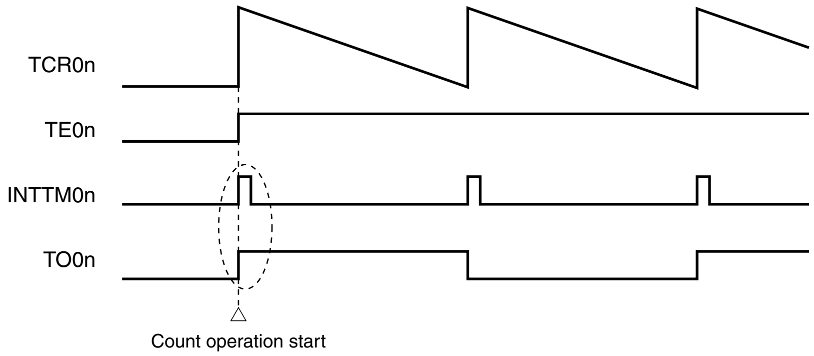
When MD0n0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTM0n) generation.

In the other modes, neither timer interrupt at count operation start nor TO0n output is controlled.

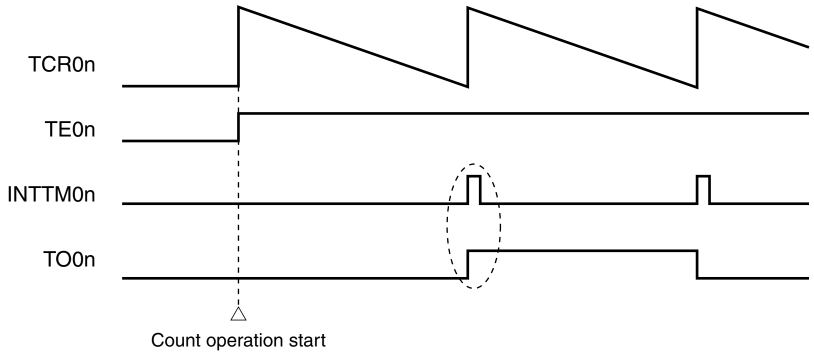
Figure 6-37 shows operation examples when the interval timer mode (TOE0n = 1, TOM0n = 0) is set.

Figure 6-37. Operation Examples of Timer Interrupt at Count Operation Start and TO0n Output

(a) When MD0n0 is set to 1



(b) When MD0n0 is set to 0



When MD0n0 is set to 1, a timer interrupt (INTTM0n) is output at count operation start, and TO0n performs a toggle operation.

When MD0n0 is set to 0, a timer interrupt (INTTM0n) is not output at count operation start, and TO0n does not change either. After counting one cycle, INTTM0n is output and TO0n performs a toggle operation.

Remark n: Channel number (n = 0, 1)

6.7 Independent Channel Operation Function of Timer Array Unit

6.7.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTM0n (timer interrupt) at fixed intervals.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTM0n (timer interrupt) = Period of count clock × (Set value of TDR0n + 1)

(2) Operation as square wave output

TO0n performs a toggle operation as soon as INTTM0n has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TO0n can be calculated by the following expressions.

- Period of square wave output from TO0n = Period of count clock × (Set value of TDR0n + 1) × 2
- Frequency of square wave output from TO0n = Frequency of count clock/{(Set value of TDR0n + 1) × 2}

Timer count register 0n (TCR0n) operates as a down counter in the interval timer mode.

The TCR0n register loads the value of timer data register 0n (TDR0n) at the first count clock after the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1. If the MD0n0 bit of timer mode register 0n (TMR0n) is 0 at this time, INTTM0n is not output and TO0n is not toggled. If the MD0n0 bit of the TMR0n register is 1, INTTM0n is output and TO0n is toggled.

After that, the TCR0n register count down in synchronization with the count clock.

When TCR0n = 0000H, INTTM0n is output and TO0n is toggled at the next count clock. At the same time, the TCR0n register loads the value of the TDR0n register again. After that, the same operation is repeated.

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid from the next period.

Remark n: Channel number (n = 0, 1).

Figure 6-38. Block Diagram of Operation as Interval Timer/Square Wave Output

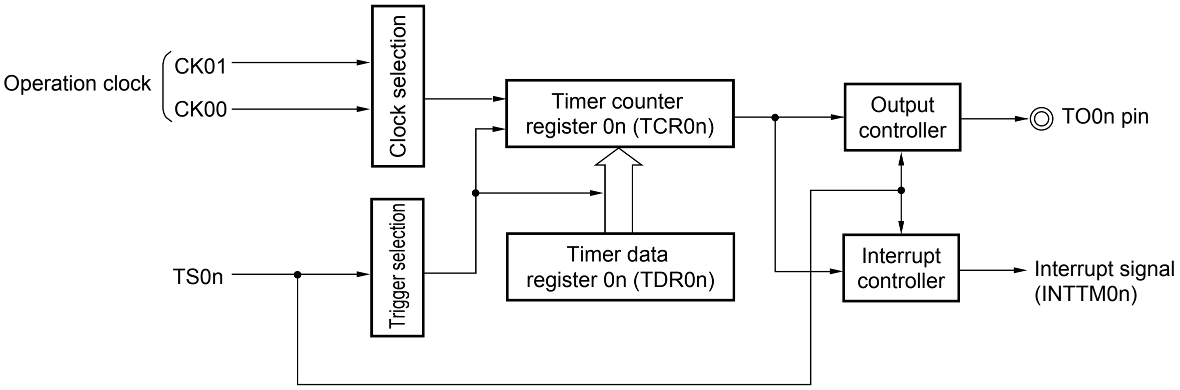
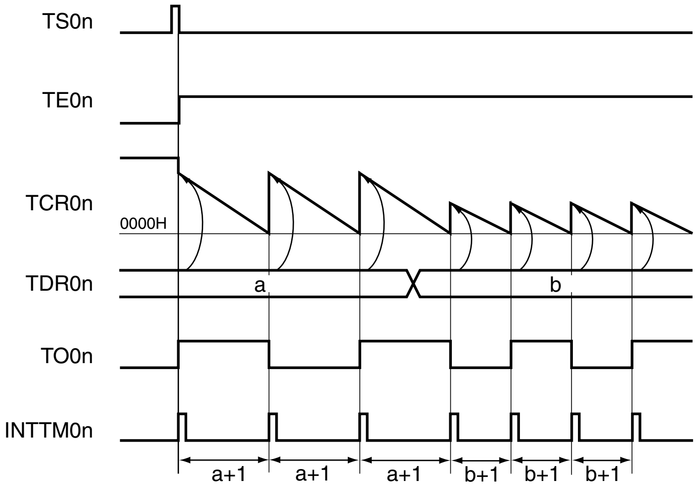


Figure 6-39. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MD0n0 = 1)



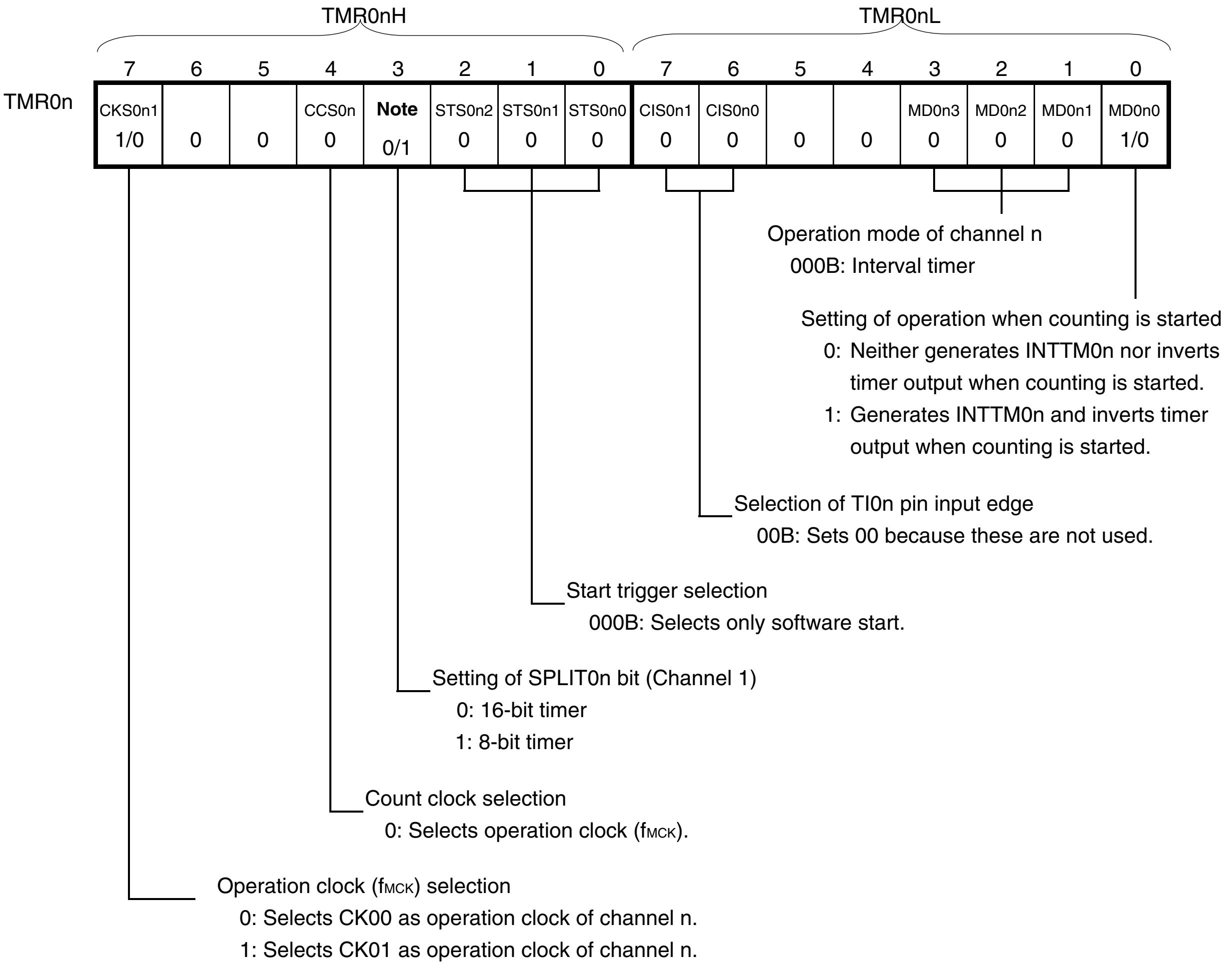
Remarks **1.** n: Channel number (n = 0, 1).

2.

TS0n:	Bit n of timer channel start register 0 (TS0)
TE0n:	Bit n of timer channel enable status register 0 (TE0)
TCR0n:	Timer count register 0n (TCR0n)
TDR0n:	Timer data register 0n (TDR0n)
TO0n:	TO0n pin output signal

Figure 6-40. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



	TMR0nH								TMR0nL							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
TMR0n	CKS0n1 1/0	0	0	CCS0n 0	**Note** 0/1	STS0n2 0	STS0n1 0	STS0n0 0	CIS0n1 0	CIS0n0 0	0	0	MD0n3 0	MD0n2 0	MD0n1 0	MD0n0 1/0

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTM0n nor inverts timer output when counting is started.
1: Generates INTTM0n and inverts timer output when counting is started.

Selection of TI0n pin input edge
00B: Sets 00 because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of SPLIT0n bit (Channel 1)
0: 16-bit timer
1: 8-bit timer

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n
TO0	TO0n 1/0

0: Outputs 0 from TO0n.
1: Outputs 1 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOE0n 1/0

0: Stops the TO0n output operation by counting operation.
1: Enables the TO0n output operation by counting operation.

Note TMR01: SPLIT01 bit
TMR00: 0 fixed

Remark n: Channel number (n = 0, 1).

Figure 6-40. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when master channel output mode (TOM0n = 0)

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Remark n: Channel number (n = 0, 1).

Figure 6-41. 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 (when the TAU0EN bit is 0, read/write operation is disabled).	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Sets interval (period) value to timer data register 0n (TDR0n).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TO0n output Clears the TOM0n bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL0n bit to 0. Sets the TO0n bit and determines default level of the TO0n output.	The TO0n pin goes into Hi-Z output state. The TO0n default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets the TOE0n bit to 1 and enables operation of TO0n. Clears the port register and port mode register to 0.	TO0n does not change because channel stops operating. The TO0n pin outputs the TO0n set level.
Operation start	(Sets the TOE0n bit to 1 only if using TO0n output and resuming operation.). Sets the TS0n (TSH01) bit to 1. The TS0n (TSH01) bit automatically returns to 0 because it is a trigger bit.	TE0n (TEH01) = 1, and count operation starts. Value of the TDR0n register is loaded to timer count register 0n (TCR0n) at the count clock input. INTTM0n is generated and TO0n performs toggle operation if the MD0n0 bit of the TMR0nL register is 1.
During operation	Set values of the TMR0n register, TOM0n, and TOL0n bits cannot be changed. Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TO0 and TOE0 registers can be changed.	Counter (TCR0n) counts down. When count value reaches 0000H, the value of the TDR0n register is loaded to the TCR0n register again and the count operation is continued. By detecting TCR0n = 0000H, INTTM0n is generated and TO0n performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTH0n (TTH01) bit is set to 1. The TTH0n (TTH01) bit automatically returns to 0 because it is a trigger bit.	TEH0n (TEH01) = 0, and count operation stops. The TCR0n register holds count value and stops. The TO0n output is not initialized but holds current status.
	The TOE0n bit is cleared to 0 and value is set to the TO0n bit.	The TO0n pin outputs the TO0n bit set level.

Operation is resumed.

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

Figure 6-41. Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TO0n pin output level Clears the TO0n bit to 0 after the value to be held is set to the port register. → When holding the TO0n pin output level is not necessary Setting not required.	The TO0n 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 TO0n bit is cleared to 0 and the TO0n pin is set to port mode.)

Remark n: Channel number (n = 0, 1).

<R> **Caution When channel 1 is used in 8-bit timer mode (SPLIT = 1), it is prohibited to read the TCR01H and TDR01H registers.**

6.7.2 Operation as external event counter

The timer array unit can be used as an external event counter that counts the number of times the valid input edge (external event) is detected in the TI0n pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDR0n + 1

Timer count register 0n (TCR0n) operates as a down counter in the event counter mode.

The TCR0n register loads the value of timer data register 0n (TDR0n) by setting any channel start trigger bit (TS0n, TSH01) of timer channel start register 0 (TS0) to 1.

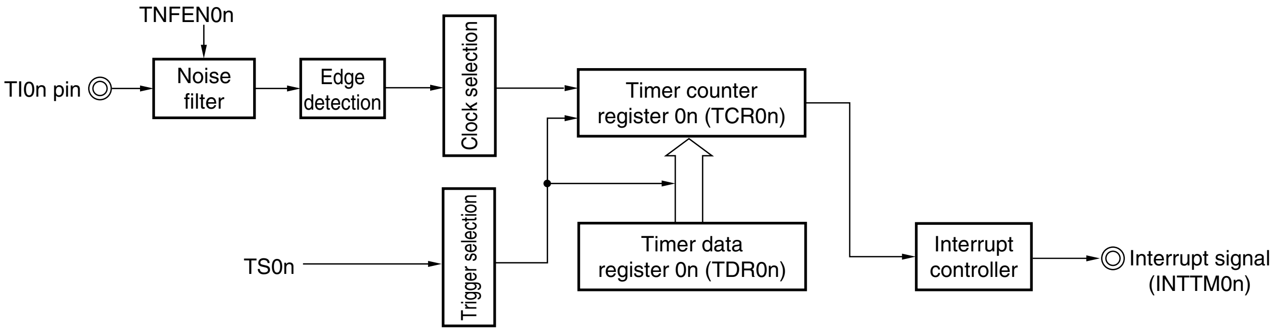
The TCR0n register counts down each time the valid input edge of the TI0n pin has been detected. When TCR0n = 0000H, the TCR0n register loads the value of the TDR0n register again, and outputs INTTM0n.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TO0n pin. Stop the output by setting the TOE0n bit of timer output enable register 0 (TOE0) to 0.

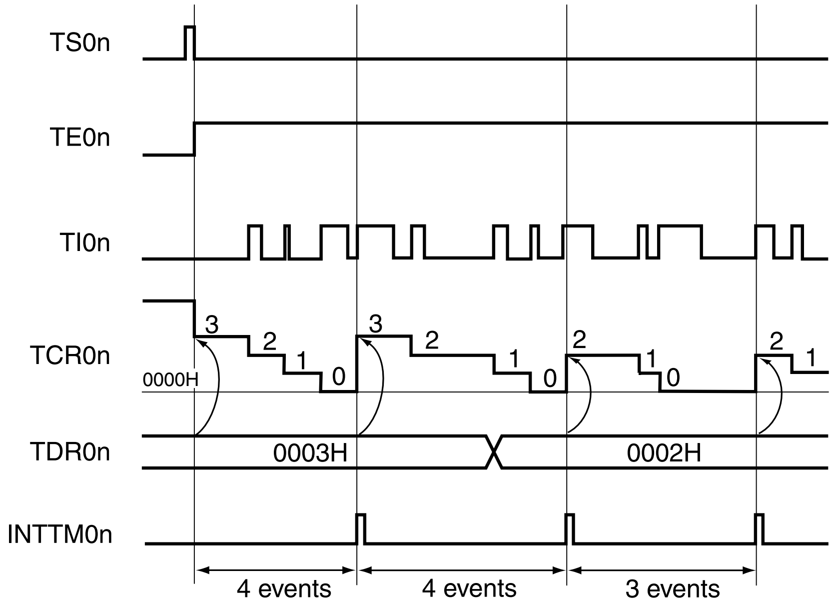
The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid during the next count period.

Figure 6-42. Block Diagram of Operation as External Event Counter



Remark n: Channel number (n = 0, 1).

Figure 6-43. Example of Basic Timing of Operation as External Event Counter



Remarks 1. n: Channel number (n = 0, 1).

2. TS0n: Bit n of timer channel start register 0 (TS0)

TE0n: Bit n of timer channel enable status register 0 (TE0)

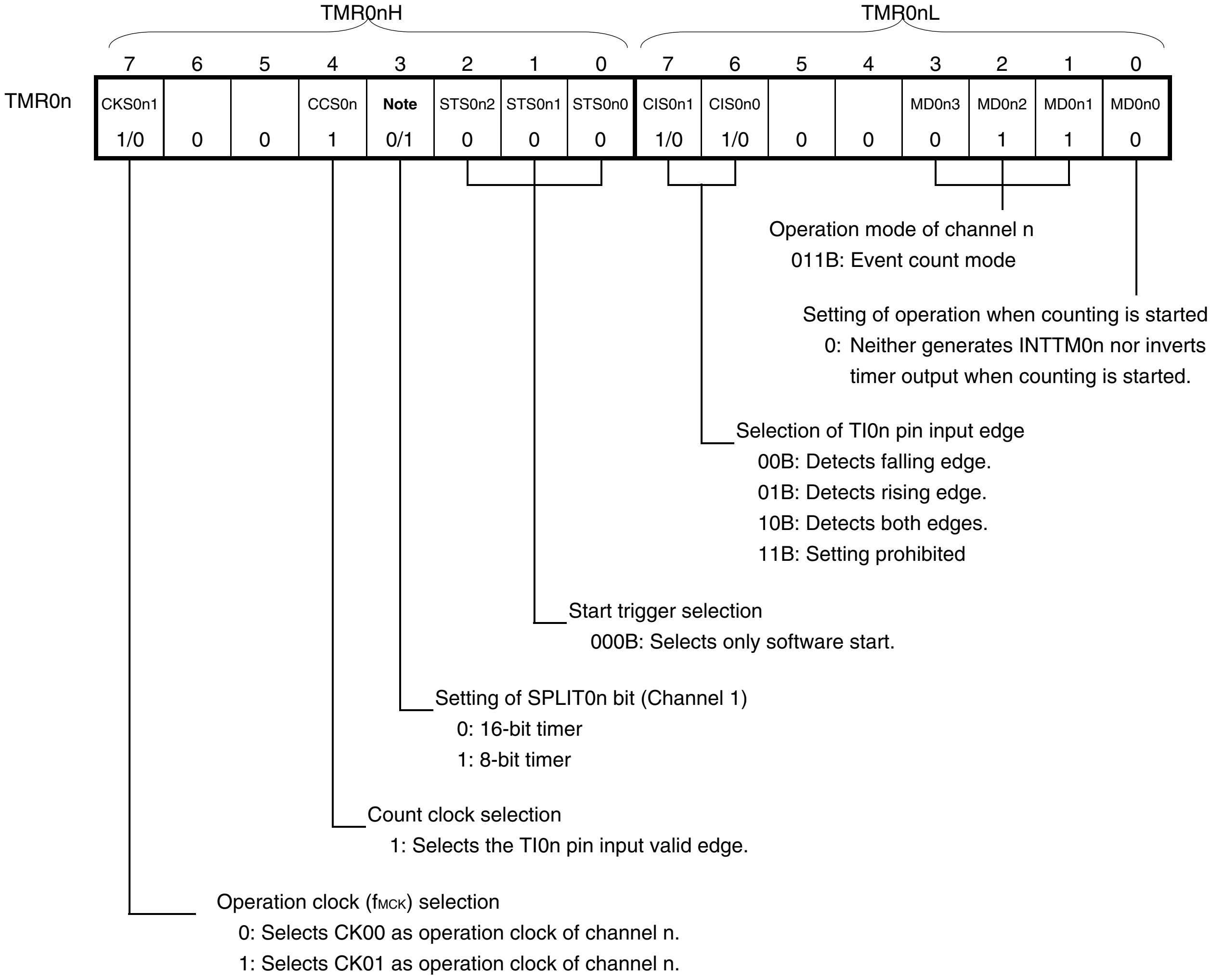
TI0n: TI0n pin input signal

TCR0n: Timer count register 0n (TCR0n)

TDR0n: Timer data register 0n (TDR0n)

Figure 6-44. Example of Set Contents of Registers in External Event Counter Mode (1/2)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
TMR0n	CKS0n1			CCS0n	**Note**	STS0n2	STS0n1	STS0n0	CIS0n1	CIS0n0			MD0n3	MD0n2	MD0n1	MD0n0
	1/0	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 INTTM0n nor inverts timer output when counting is started.

Selection of TI0n 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 SPLIT0n bit (Channel 1)
0: 16-bit timer
1: 8-bit timer

Count clock selection
1: Selects the TI0n pin input valid edge.

Operation clock (f_{MCK}) selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

Note TMR01: SPLIT01 bit
TMR00: 0 fixed

Remark n: Channel number (n = 0, 1).

Figure 6-44. Example of Set Contents of Registers in External Event Counter Mode (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n	0: Cleared to 0 when master channel output mode (TOM0n = 0).
	0	

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n	0: Sets master channel output mode.
	0	

Remark n: Channel number (n = 0, 1).

Figure 6-45. 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 (when the TAU0EN bit is 0, read/write operation is disabled). →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Sets number of counts to timer data register 0n (TDR0n). Sets noise filter enable register 1 (NFEN1) Clears the TOE0n bit of timer output enable register 0 (TOE0) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and count operation starts. Value of the TDR0n register is loaded to timer count register 0n (TCR0n) and detection of the TI0n pin input edge is awaited.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TMR0n register, TOM0n, TOL0n, TO0n, and TOE0n bits cannot be changed.	Counter (TCR0n) counts down each time input edge of the TI0n pin has been detected. When count value reaches 0000H, the value of the TDR0n register is loaded to the TCR0n register again, and the count operation is continued. By detecting TCR0n = 0000H, the INTTM0n output is generated. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. The TCR0n register holds count value and stops.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 0, 1).

6.7.3 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 6-46. Block Diagram of Operation as Frequency Divider

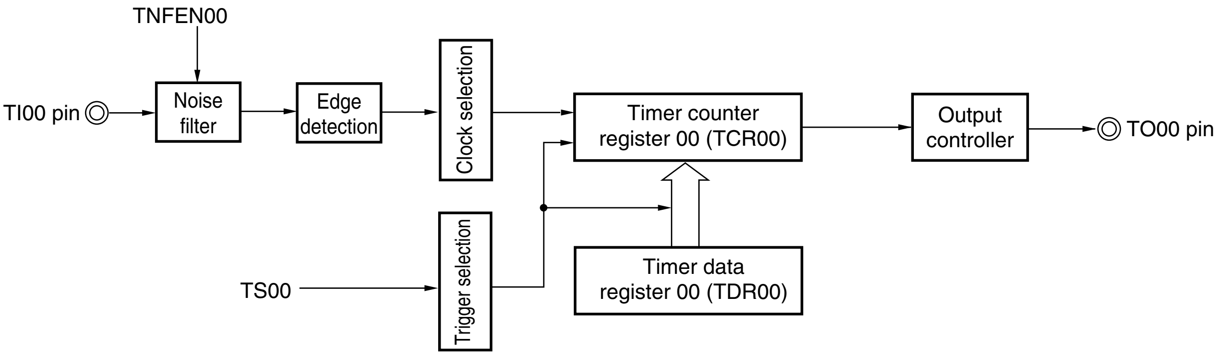
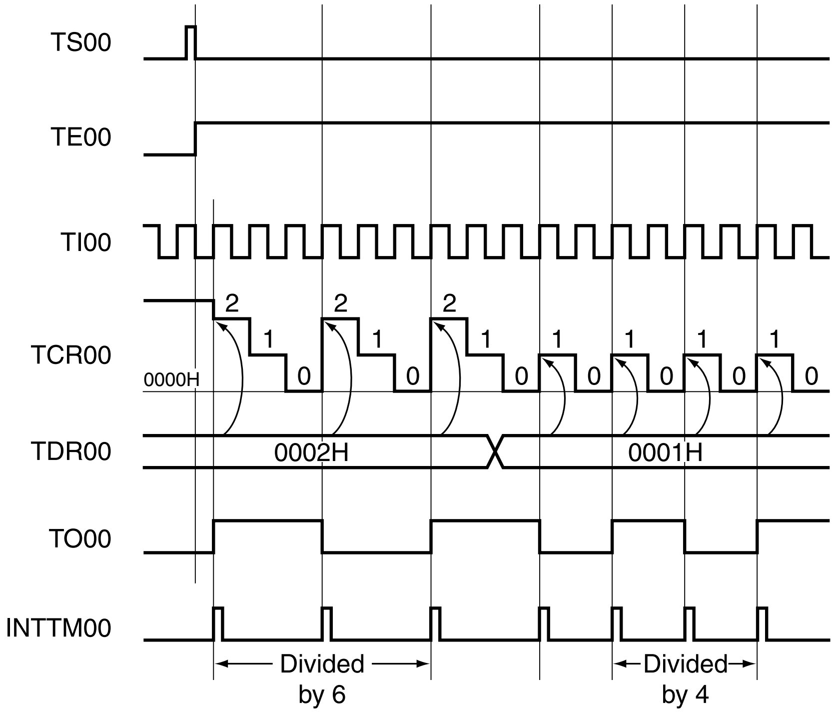


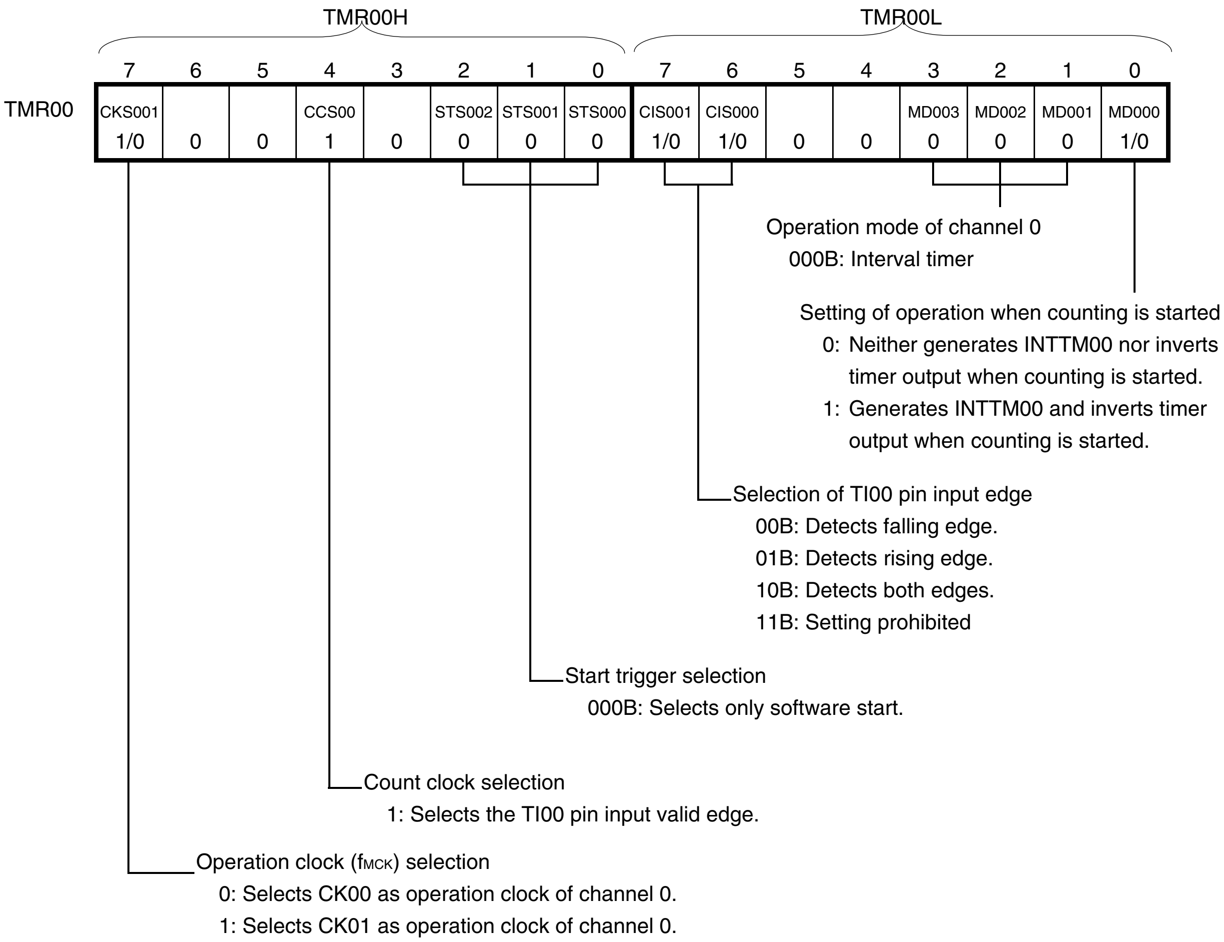
Figure 6-47. Example of Basic Timing of Operation as Frequency Divider (MD000 = 1)



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 6-48. Example of Set Contents of Registers During Operation as Frequency Divider

(a) Timer mode register 00 (TMR00H, TMR00L)



	TMR00H								TMR00L							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
TMR00	CKS001			CCS00		STS002	STS001	STS000	CIS001	CIS000			MD003	MD002	MD001	MD000
	1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	0	0	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.

Count clock selection
1: Selects the TI00 pin input valid edge.

Operation clock (fMCK) selection
0: Selects CK00 as operation clock of channel 0.
1: 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 6-49. 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 (when the TAU0EN bit is 0, read/write operation is disabled).	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel and selects the detection edge). Sets interval (period) value to timer data register 00 (TDR00). Sets noise filter enable register 1 (NFEN1)	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOM00 bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL00 bit to 0. Sets the TO00 bit and determines default level of the TO00 output. Sets the TOE00 bit to 1 and enables operation of TO00. Clears the port register and port mode register to 0.	The TO00 pin goes into Hi-Z output state. The TO00 default setting level is output when the port mode register is in output mode and the port register is 0. TO00 does not change because channel stops operating. The TO00 pin outputs the TO00 set level.
Operation start	Sets the TOE00 bit to 1 (only when operation is resumed). Sets the TS00 bit to 1. The TS00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 1, and count operation starts. Value of the TDR00 register is loaded to timer count register 00 (TCR00) at the count clock input. INTTM00 is generated and TO00 performs toggle operation if the MD000 bit of the TMR00 register is 1.
During operation	Set value of the TDR00 register can be changed. The TCR00 register can always be read. The TSR00 register is not used. Set values of the TO0 and TOE0 registers can be changed. Set values of the TMR00 register, TOM00, and TOL00 bits cannot be changed.	Counter (TCR00) counts down. When count value reaches 0000H, the value of the TDR00 register is loaded to the TCR00 register again, and the count operation is continued. By detecting TCR00 = 0000H, INTTM00 is generated and TO00 performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT00 bit is set to 1. The TT00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 0, and count operation stops. The TCR00 register holds count value and stops. The TO00 output is not initialized but holds current status.
	The TOE00 bit is cleared to 0 and value is set to the TO00 bit.	The TO00 pin outputs the TO00 set level.
TAU stop	To hold the TO00 pin output level Clears the TO00 bit to 0 after the value to be held is set to the port register. When holding the TO00 pin output level is not necessary Setting not required.	The TO00 pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0.	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.

6.7.4 Operation as input pulse interval measurement

The count value can be captured at the TI0n valid edge and the interval of the pulse input to TI0n can be measured.

The pulse interval can be calculated by the following expression.

TI0n input pulse interval = Period of count clock × ((10000H × TSR0n: OVF) + (Capture value of TDR0n + 1))

Caution The TI0n pin input is sampled using the operating clock selected with the CKS0n1 bit of timer mode register 0n (TMR0n), so an error of up to one operating clock cycle occurs.

Timer count register 0n (TCR0n) operates as an up counter in the capture mode.

When the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the TCR0n register counts up from 0000H in synchronization with the count clock.

When the TI0n pin input valid edge is detected, the count value of the TCR0n register is transferred (captured) to timer data register 0n (TDR0n) and, at the same time, the TCR0n register is cleared to 0000H, and the INTTM0n is output. If the counter overflows at this time, the OVF bit of timer status register 0n (TSR0n) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

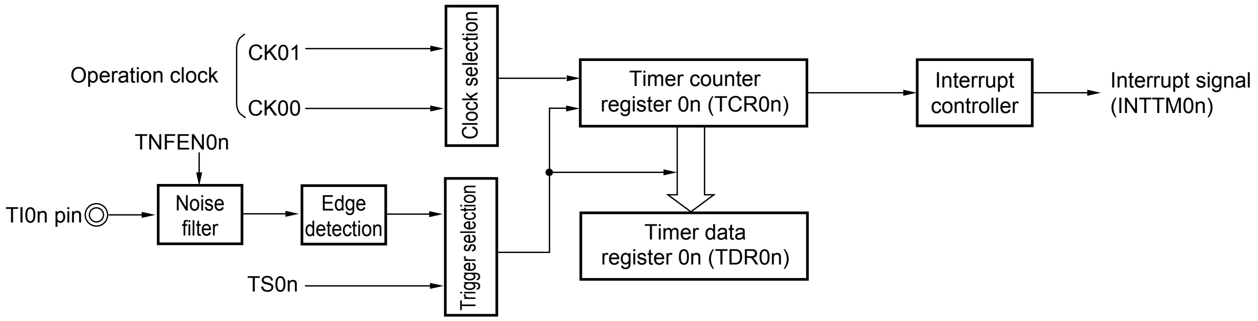
As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Set the STS0n2 to STS0n0 bits of the TMR0n register to 001B to use the valid edges of TI0n as a start trigger and a capture trigger.

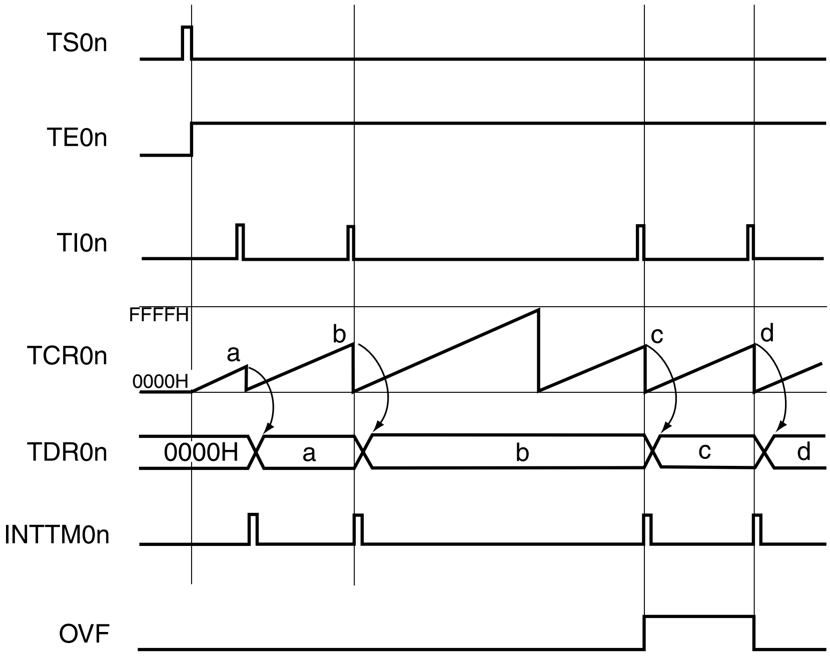
When TE0n = 1, a software operation (TS0n = 1) can be used as a capture trigger, instead of using the TI0n pin input.

Figure 6-50. Block Diagram of Operation as Input Pulse Interval Measurement



Remark n: Channel number (n = 0, 1).

Figure 6-51. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MD0n0 = 0)



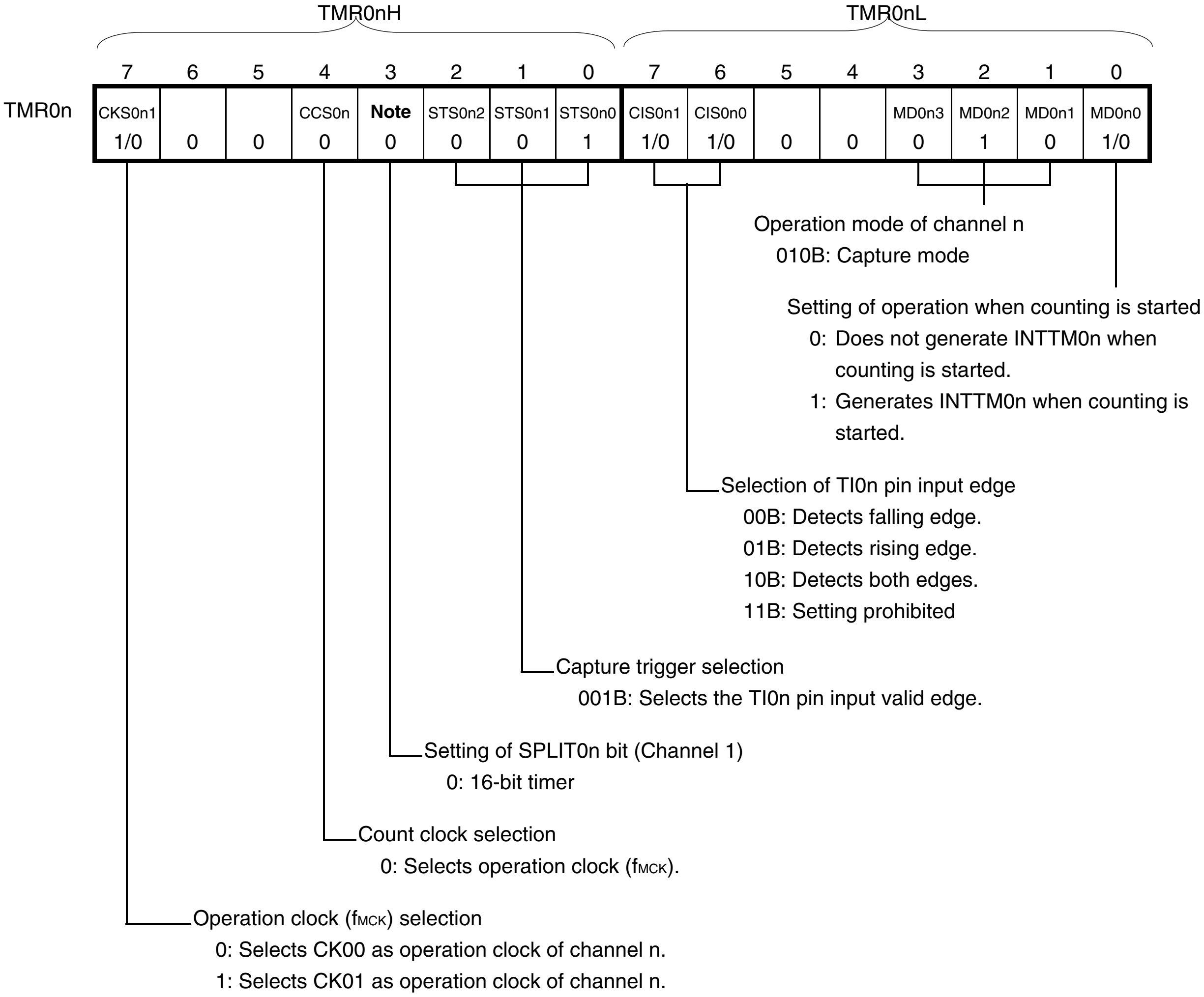
Remarks **1.** n: Channel number (n = 0, 1).

2.

TS0n:	Bit n of timer channel start register 0 (TS0)
TE0n:	Bit n of timer channel enable status register 0 (TE0)
TI0n:	TI0n pin input signal
TCR0n:	Timer count register 0n (TCR0n)
TDR0n:	Timer data register 0n (TDR0n)
OVF:	Bit 0 of timer status register 0n (TSR0n)

Figure 6-52. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register 0n (TMR0nH, TMR0nL)



	TMR0nH								TMR0nL							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
TMR0n	CKS0n1 1/0	0	0	CCS0n 0	**Note** 0	STS0n2 0	STS0n1 0	STS0n0 1	CIS0n1 1/0	CIS0n0 1/0	0	0	MD0n3 0	MD0n2 1	MD0n1 0	MD0n0 1/0

Operation mode of channel n
010B: Capture mode

Setting of operation when counting is started
0: Does not generate INTTM0n when counting is started.
1: Generates INTTM0n when counting is started.

Selection of TI0n pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Capture trigger selection
001B: Selects the TI0n pin input valid edge.

Setting of SPLIT0n bit (Channel 1)
0: 16-bit timer

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops TO0n output operation by counting operation.

Note TMR01: SPLIT01 bit
TMR00: 0 fixed

Remark n: Channel number (n = 0, 1).

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n	0: Cleared to 0 when master channel output mode (TOM0n = 0).
	0	

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n	0: Sets master channel output mode.
	0	

Remark n: Channel number (n = 0, 1).

Figure 6-53. 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 (when the TAU0EN bit is 0, read/write operation is disabled). →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Sets Noise filter enable register 1 (NFEN1).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and count operation starts. Timer count register 0n (TCR0n) is cleared to 0000H at the count clock input. When the MD0n0 bit of the TMR0n register is 1, INTTM0n is generated.
During operation	Set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. The TDR0n register can always be read. The TCR0n register can always be read. The TSR0n register can always be read. Set values of the TOM0n, TOL0n, TO0n, and TOE0n bits cannot be changed.	Counter (TCR0n) counts up from 0000H. When the TI0n pin input valid edge is detected, the count value is transferred (captured) to timer data register 0n (TDR0n). At the same time, the TCR0n register is cleared to 0000H, and the INTTM0n signal is generated. If an overflow occurs at this time, the OVF bit of timer status register 0n (TSR0n) is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. The TCR0n register holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 0, 1).

6.7.5 Operation as input signal high-/low-level width measurement

By starting counting at one edge of the TI0n pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TI0n can be measured. The signal width of TI0n can be calculated by the following expression.

Signal width of TI0n input = Period of count clock × ((10000H × TSR0n: OVF) + (Capture value of TDR0n + 1))

Caution The TI0n pin input is sampled using the operating clock selected with the CKS0n1 bit of timer mode register 0n (TMR0n), so an error equivalent to one operation clock occurs.

Timer count register 0n (TCR0n) operates as an up counter in the capture & one-count mode.

When the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the TE0n bit is set to 1 and the TI0n pin start edge detection wait status is set.

When the TI0n pin input start edge (rising edge of the TI0n pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TI0n pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register 0n (TDR0n) and, at the same time, INTTM0n is output. If the counter overflows at this time, the OVF bit of timer status register 0n (TSR0n) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCR0n register stops at the value “value transferred to the TDR0n register + 1”, and the TI0n pin start edge detection wait status is set. After that, the above operation is repeated.

As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Whether the high-level width or low-level width of the TI0n pin is to be measured can be selected by using the CIS0n1 and CIS0n0 bits of the TMR0n register.

Because this function is used to measure the signal width of the TI0n pin input, the TS0n bit cannot be set to 1 while the TE0n bit is 1.

CIS0n1, CIS0n0 of TMR0n register = 10B: Low-level width is measured.
CIS0n1, CIS0n0 of TMR0n register = 11B: High-level width is measured.

Figure 6-54. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement

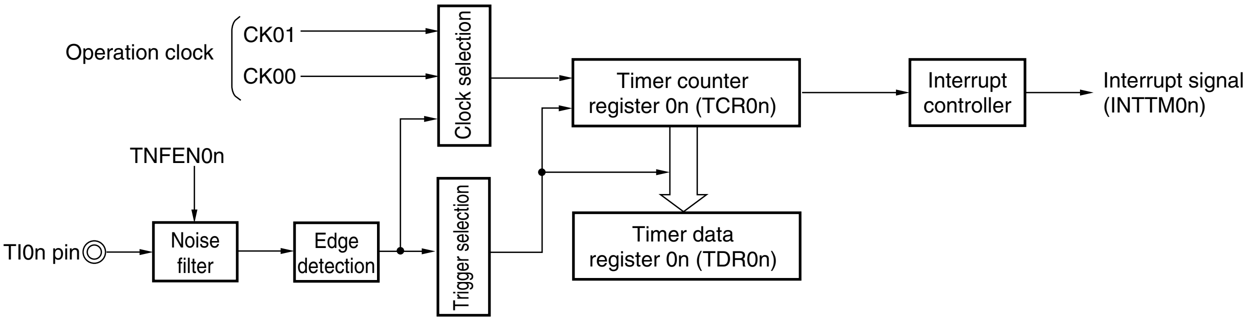
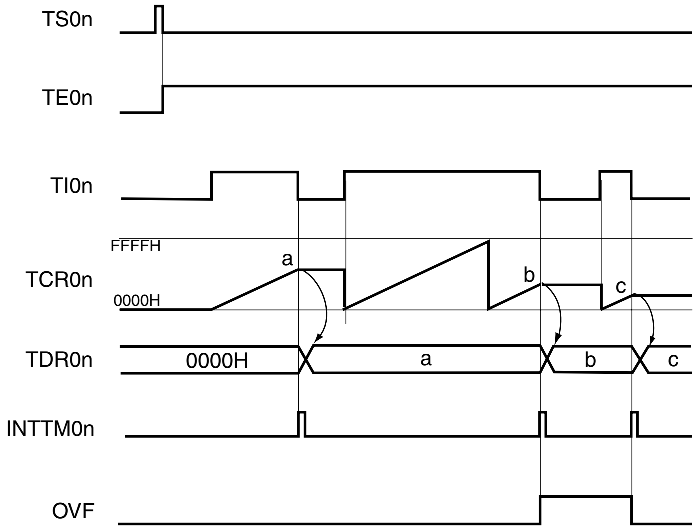


Figure 6-55. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remarks **1.** n: Channel number (n = 0, 1).

2.

TS0n:	Bit n of timer channel start register 0 (TS0)
TE0n:	Bit n of timer channel enable status register 0 (TE0)
TI0n:	TI0n pin input signal
TCR0n:	Timer count register 0n (TCR0n)
TDR0n:	Timer data register 0n (TDR0n)
OVF:	Bit 0 of timer status register 0n (TSR0n)

Figure 6-56. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register 0n (TMR0nH, TMR0nL)

	TMR0nH								TMR0nL							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
TMR0n	CKS0n1			CCS0n	**Note**	STS0n2	STS0n1	STS0n0	CIS0n1	CIS0n0			MD0n3	MD0n2	MD0n1	MD0n0
	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 INTTM0n when counting is started.

Selection of TI0n 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 TI0n pin input valid edge.

Setting of SPLIT0n bit (Channel 1)
0: 16-bit timer

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when master channel output mode (TOM0n = 0).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Note TMR01: SPLIT01 bit
TMR00: 0 fixed

Remark n: Channel number (n = 0, 1).

Figure 6-57. 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 (when the TAU0EN bit is 0, read/write operation is disabled). →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel). Sets noise filter enable register 1 (NFEN1) Clears the TOE0n bit to 0 and stops operation of TO0n.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and the TI0n pin start edge detection wait status is set.
	Detects the TI0n pin input count start valid edge. →	Clears timer count register 0n (TCR0n) to 0000H and starts counting up.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TMR0n register, TOM0n, TOL0n, TO0n, and TOE0n bits cannot be changed.	When the TI0n pin start edge is detected, the counter (TCR0n) counts up from 0000H. If a capture edge of the TI0n pin is detected, the count value is transferred to timer data register 0n (TDR0n) and INTTM0n is generated. If an overflow occurs at this time, the OVF bit of timer status register 0n (TSR0n) is set; if an overflow does not occur, the OVF bit is cleared. The TCR0n register stops the count operation until the next TI0n pin start edge is detected.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. The TCR0n register holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 0, 1).

6.7.6 Operation as delay counter

It is possible to start counting down when the valid edge of the TI0n pin input is detected (an external event), and then generate INTTM0n (a timer interrupt) after any specified interval.

It can also generate INTTM0n (timer interrupt) at any interval by making a software set TS0n = 1 and the count down start during the period of TE0n = 1.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTM0n (timer interrupt) = Period of count clock × (Set value of TDR0n + 1)

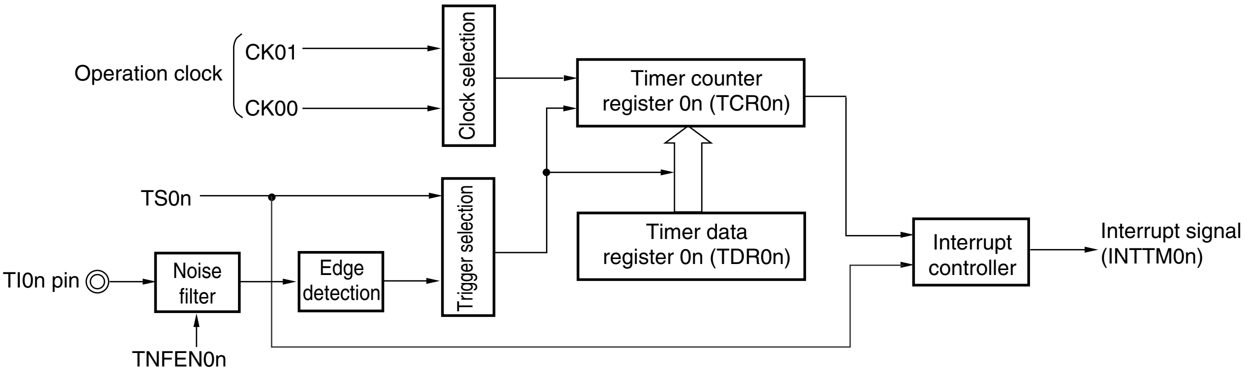
Timer count register 0n (TCR0n) operates as a down counter in the one-count mode.

When the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the TE0n bit is set to 1 and the TI0n pin input valid edge detection wait status is set.

Timer count register 0n (TCR0n) starts operating upon TI0n pin input valid edge detection and loads the value of timer data register 0n (TDR0n). The TCR0n register counts down from the value of the TDR0n register it has loaded, in synchronization with the count clock. When TCR0n = 0000H, it outputs INTTM0n and stops counting until the next TI0n pin input valid edge is detected.

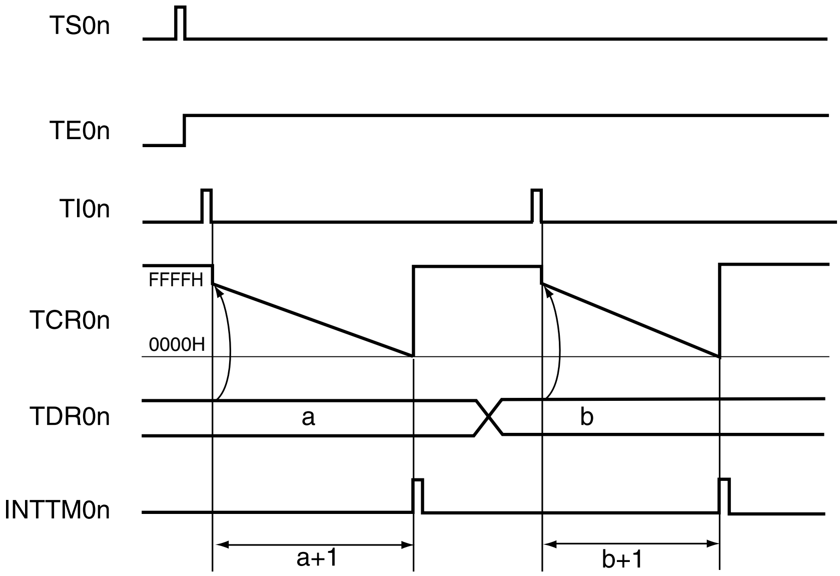
The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid from the next period.

Figure 6-58. Block Diagram of Operation as Delay Counter



Remark n: Channel number (n = 0, 1).

Figure 6-59. Example of Basic Timing of Operation as Delay Counter



Remarks 1. n: Channel number (n = 0, 1).

2. TS0n: Bit n of timer channel start register 0 (TS0)

TE0n: Bit n of timer channel enable status register 0 (TE0)

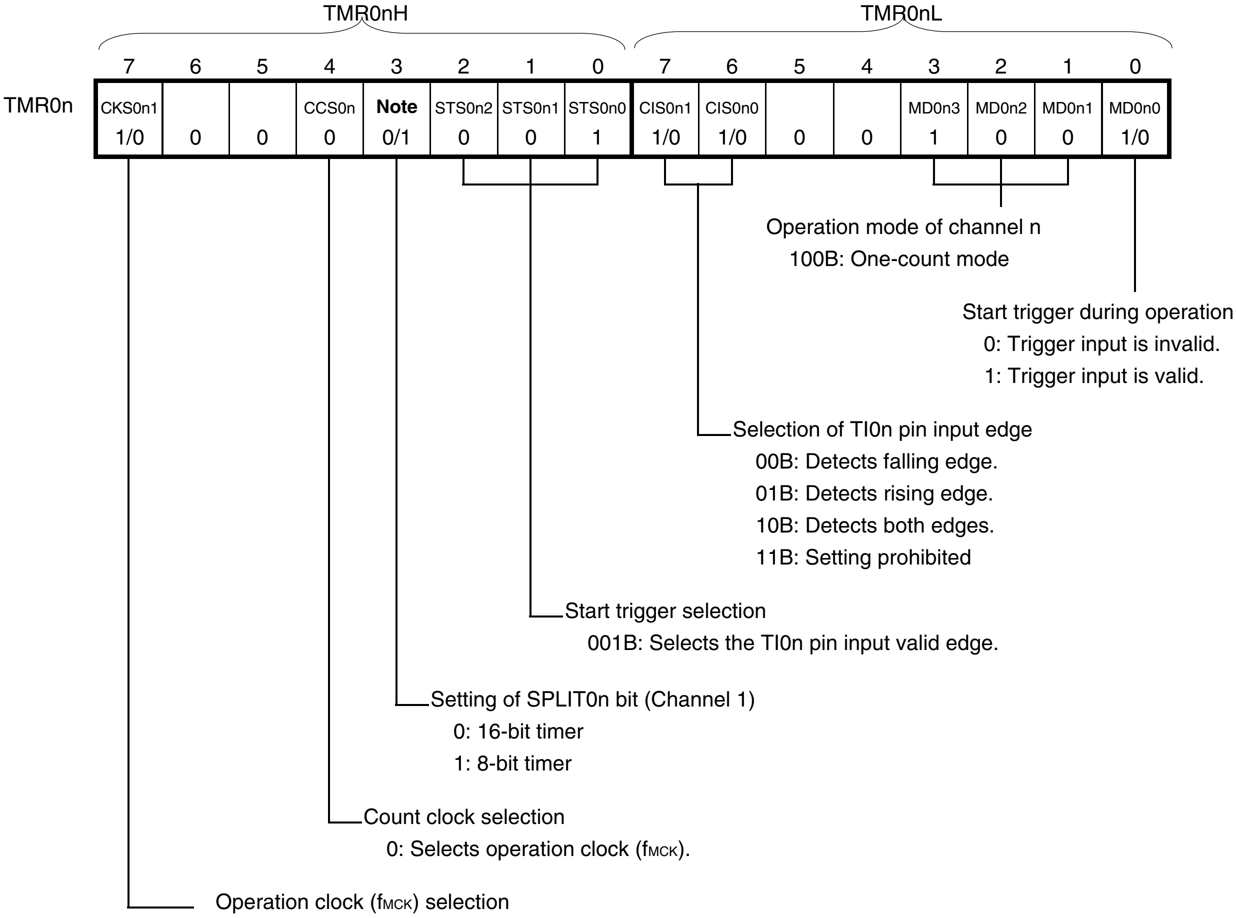
TI0n: TI0n pin input signal

TCR0n: Timer count register 0n (TCR0n)

TDR0n: Timer data register 0n (TDR0n)

Figure 6-60. Example of Set Contents of Registers to Delay Counter (1/2)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



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 TI0n pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TI0n pin input valid edge.

Setting of SPLIT0n bit (Channel 1)
0: 16-bit timer
1: 8-bit timer

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK02 as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

Note TMR01: SPLIT01 bit
TMR00: 0 fixed

Remark n: Channel number (n = 0, 1).

Figure 6-60. Example of Set Contents of Registers to Delay Counter (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when master channel output mode (TOM0n = 0).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Remark n: Channel number (n = 0, 1).

Figure 6-61. 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 (when the TAU0EN bit is 0, read/write operation is disabled).	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel). INTTM0n output delay is set to timer data register 0n (TDR0n). Sets noise filter enable register 1 (NFEN1). Clears the TOE0n bit to 0 and stops operation of TO0n.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0n bit to 1. The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and the TI0n pin input valid edge detection wait status is set.
	Detects the TI0n pin input valid edge.	Value of the TDR0n register is loaded to the timer count register 0n (TCR0n).
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used.	The counter (TCR0n) counts down. When TCR0n counts down to 0000H, INTTM0n is output, and counting stops (which leaves TCR0n at FFFFH) until the next TI0n pin input. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. The TCR0n register holds count value and stops.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n: Channel number (n = 0, 1).

6.8 Simultaneous Channel Operation Function of Timer Array Unit

6.8.1 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 TI0n pin.

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDR0n (master) + 2} × Count clock period
Pulse width = {Set value of TDR0p (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer count register 0n (TCR0n) of the master channel starts operating upon start trigger detection and loads the value of timer data register 0n (TDR0n).

The TCR0n register counts down from the value of the TDR0n register it has loaded, in synchronization with the count clock. When TCR0n = 0000H, it outputs INTTM0n and stops counting until the next start trigger is detected.

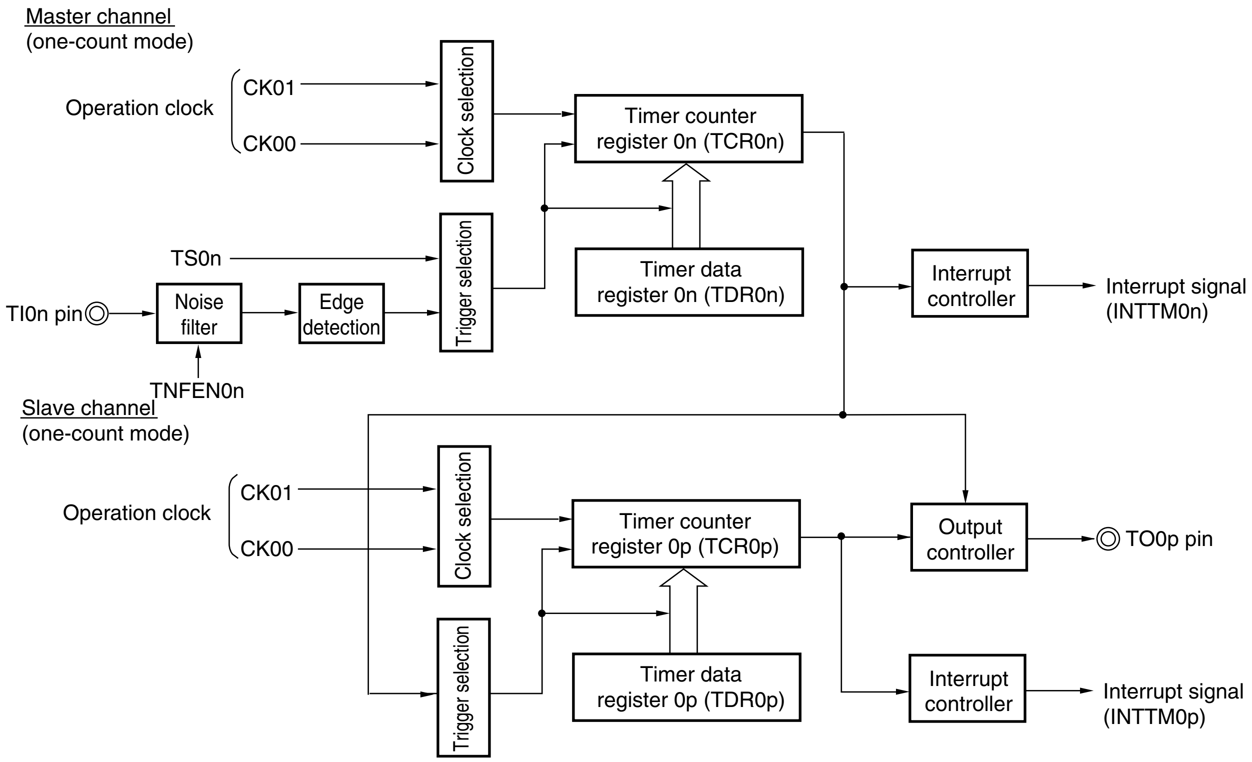
The slave channel operates in the one-count mode and counts the pulse width. The TCR0p register of the slave channel starts operation using INTTM0n of the master channel as a start trigger, and loads the value of the TDR0p register. The TCR0p register counts down from the value of The TDR0p register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTM0p and stops counting until the next start trigger (INTTM0n of the master channel) is detected. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H.

Instead of using the TI0n pin input, a one-shot pulse can also be output using the software operation (TS0n = 1) as a start trigger.

Caution The timing of loading of timer data register 0n (TDR0n) of the master channel is different from that of the TDR0p register of the slave channel. If the TDR0n and TDR0p registers are rewritten during operation, therefore, an illegal waveform is output. Rewrite the TDR0n register after INTTM0n is generated and the TDR0p register after INTTM0p is generated.

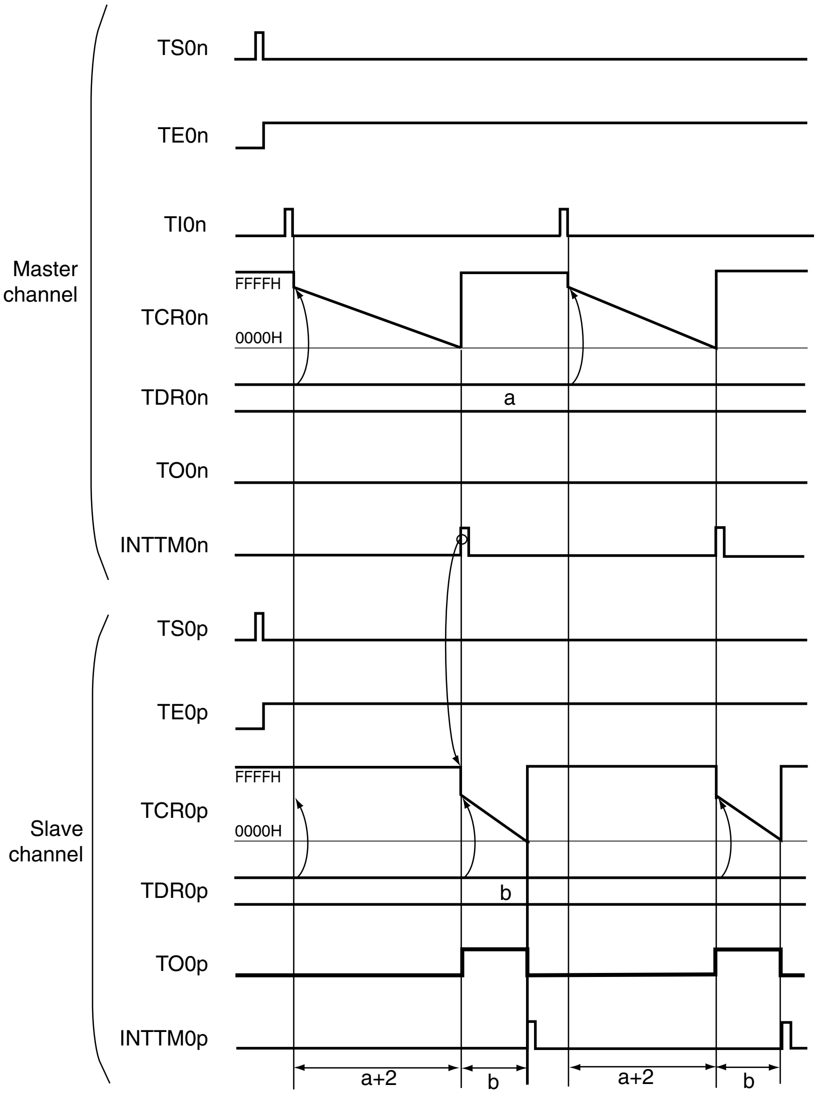
Remark n: Master channel number (n = 0)
p: Slave channel number (p = 1)

Figure 6-62. Block Diagram of Operation as One-Shot Pulse Output Function



Remark n: Master channel number (n = 0)
p: Slave channel number (p = 1)

Figure 6-63. Example of Basic Timing of Operation as One-Shot Pulse Output Function



Remarks 1. n: Master channel number (n = 0)

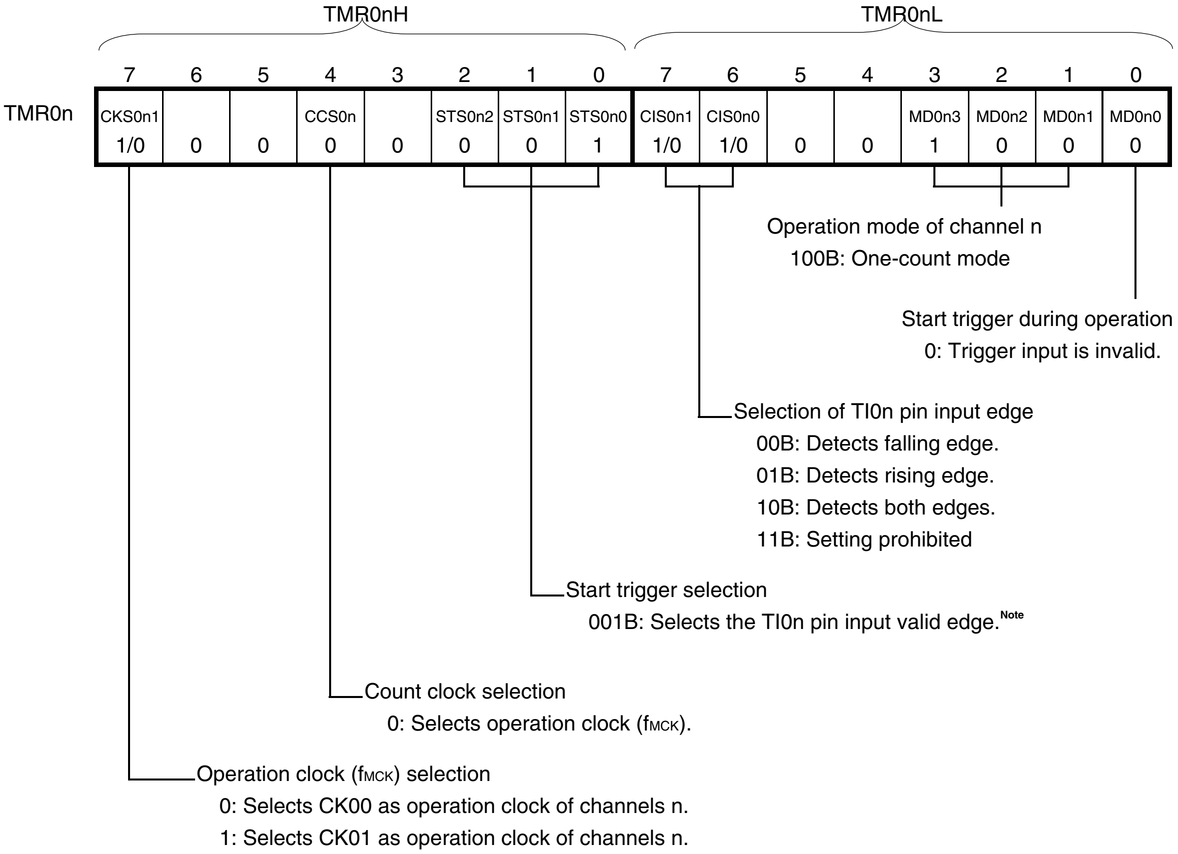
p: Slave channel number (p = 1)

2.

TS0n, TS0p:	Bit n, p of timer channel start register 0 (TS0)
TE0n, TE0p:	Bit n, p of timer channel enable status register 0 (TE0)
TI0n, TI0p:	TI0n and TI0p pins input signal
TCR0n, TCR0p:	Timer count registers 0n, 0p (TCR0n, TCR0p)
TDR0n, TDR0p:	Timer data registers 0n, 0p (TDR0n, TDR0p)
TO0n, TO0p:	TO0n and TO0p pins output signal

Figure 6-64. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



	TMR0nH								TMR0nL							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
TMR0n	CKS0n1 1/0	0	0	CCS0n 0	0	STS0n2 0	STS0n1 0	STS0n0 1	CIS0n1 1/0	CIS0n0 1/0	0	0	MD0n3 1	MD0n2 0	MD0n1 0	MD0n0 0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TI0n pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TI0n pin input valid edge.[Note]

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
0: Selects CK00 as operation clock of channels n.
1: Selects CK01 as operation clock of channels n.

(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

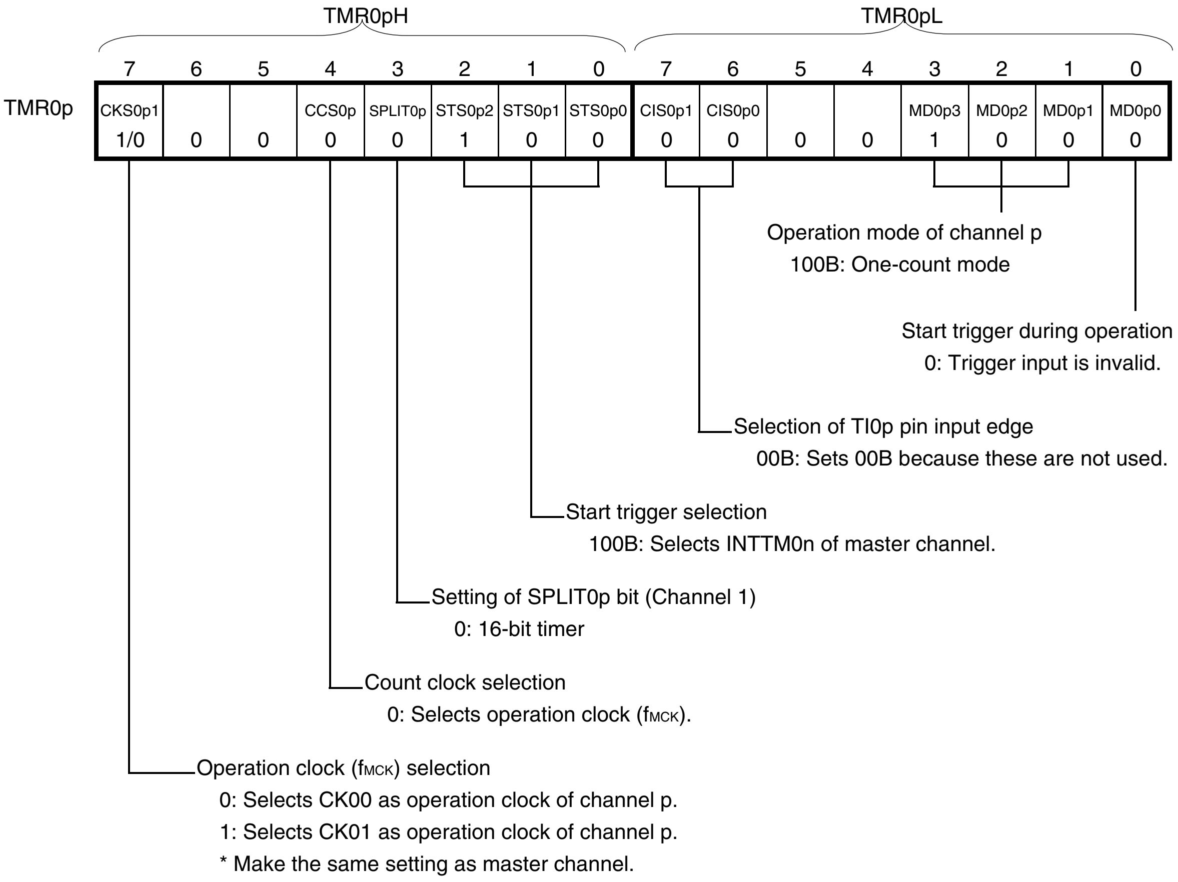
	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Note Instead of using the TI0n pin input, a one-shot pulse can also be output using the software operation (TS0n = 1) as a start trigger.

Remark n: Master channel number (n = 0)

Figure 6-65. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register 0p (TMR0pH, TMR0pL)



	TMR0pH								TMR0pL							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
TMR0p	CKS0p1 1/0	0	0	CCS0p 0	SPLIT0p 0	STS0p2 1	STS0p1 0	STS0p0 0	CIS0p1 0	CIS0p0 0	0	0	MD0p3 1	MD0p2 0	MD0p1 0	MD0p0 0

Operation mode of channel p
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TI0p pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTM0n of master channel.

Setting of SPLIT0p bit (Channel 1)
0: 16-bit timer

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
0: Selects CK00 as operation clock of channel p.
1: Selects CK01 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register 0 (TO0)

	Bit p
TO0	TO0p 1/0

0: Outputs 0 from TO0p.
1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p
TOE0	TOE0p 1/0

0: Stops the TO0p output operation by counting operation.
1: Enables the TO0p output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit p
TOL0	TOL0p 1/0

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit p
TOM0	TOM0p 1

1: Sets the slave channel output mode.

Note TMR01: SPLIT01 bit
TMR00: 0 fixed

Remark n: Master channel number (n = 0)
p: Slave channel number (p = 1)

Figure 6-66. 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 (when the TAU0EN bit is 0, read/write operation is disabled).	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode register 0n, mp (TMR0n, TMR0p) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register 0n (TDR0n) of the master channel, and a pulse width is set to the TDR0p register of the slave channel. Sets Noise filter enable register 1 (NFEN1) of the master channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0p bit of timer output mode register 0 (TOM0) is set to 1 (slave channel output mode). Sets the TOL0p bit. Sets the TO0p bit and determines default level of the TO0p output.	The TO0p pin goes into Hi-Z output state. The TO0p default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE0p bit to 1 and enables operation of TO0p.	TO0p does not change because channel stops operating.
	Clears the port register and port mode register to 0.	The TO0p pin outputs the TO0p set level.

Remark n: Master channel number (n = 0)
p: Slave channel number (p = 1)

Figure 6-66. Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
Operation start	Sets the TOE0p bit (slave) to 1 (only when operation is resumed). The TS0n (master) and TS0p (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. The TS0n and TS0p bits automatically return to 0 because they are trigger bits.	The TE0n and TE0p bits are set to 1 and the master channel enters the TI0n input edge detection wait status. Counter stops operating.
	Detects the TI0n pin input valid edge of master channel.	Master channel starts counting.
During operation	Set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. Set values of the TMR0p, TDR0n, TDR0p registers, TOM0n, TOM0p, TOL0n, and TOL0p bits cannot be changed. The TCR0n and TCR0p registers can always be read. The TSR0n and TSR0p registers are not used. Set values of the TO0p and TOE0p registers of slave channel can be changed.	Master channel loads the value of the TDR0n register to timer count register 0n (TCR0n) when the TI0n pin valid input edge is detected, and the counter starts counting down. When the count value reaches TCR0n = 0000H, the INTTM0n output is generated, and the counter stops until the next valid edge is input to the TI0n pin. The slave channel, triggered by INTTM0n of the master channel, loads the value of the TDR0p register to the TCR0p register, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n (master) and TT0p (slave) bits are set to 1 at the same time. The TT0n and TT0p bits automatically return to 0 because they are trigger bits.	TE0n, TE0p = 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized but holds current status.
	The TOE0p bit of slave channel is cleared to 0 and value is set to the TO0p bit.	The TO0p pin outputs the TO0p set level.
TAU stop	To hold the TO0p pin output level Clears the TO0p bit to 0 after the value to be held is set to the port register. When holding the TO0p pin output level is not necessary Setting not required.	The TO0p 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 TO0p bit is cleared to 0 and the TO0p pin is set to port mode.)

Operation is resumed.

Remark n: Master channel number (n = 0)

p: Slave channel number (p = 1)

6.8.2 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDR0n (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDR0p (slave)}/{Set value of TDR0n (master) + 1} × 100
0% output: Set value of TDR0p (slave) = 0000H
100% output: Set value of TDR0p (slave) ≥ {Set value of TDR0n (master) + 1}

Remark The duty factor exceeds 100% if the set value of TDR0p (slave) > (set value of TDR0n (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, an interrupt (INTTM0n) is output, the value set to timer data register 0n (TDR0n) is loaded to timer count register 0n (TCR0n), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTM0n is output, the value of the TDR0n register is loaded again to the TCR0n register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TT0n) of timer channel stop register 0 (TT0) is set to 1.

If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TO0p) cycle.

The slave channel operates in one-count mode. By using INTTM0n from the master channel as a start trigger, the TCR0p register loads the value of the TDR0p register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTM0p and waits until the next start trigger (INTTM0n from the master channel) is generated.

If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TO0p) duty.

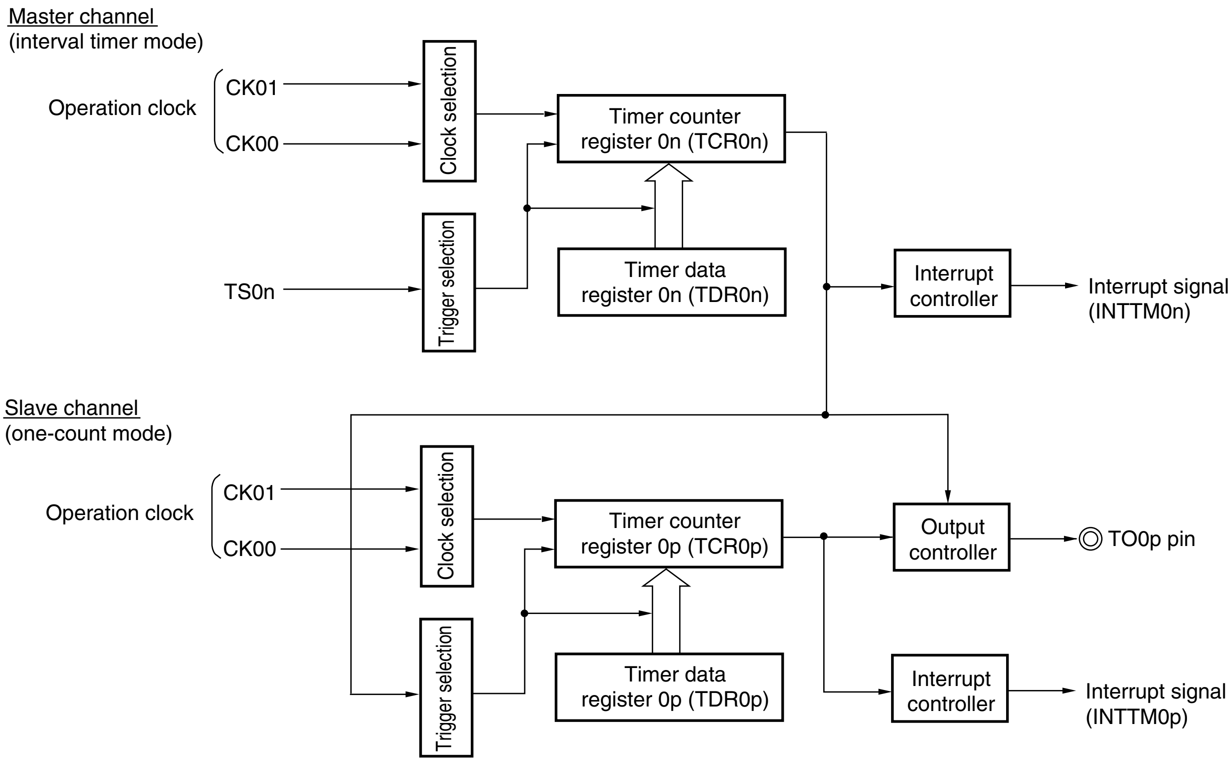
PWM output (TO0p) goes to the active level one clock after the master channel generates INTTM0n and goes to the inactive level when the TCR0p register of the slave channel becomes 0000H.

Caution 1. To rewrite both timer data register 0n (TDR0nH, TDR0nL) of the master channel and the TDR0pH and TDR0pL registers of the slave channel, a write access is necessary at least four times. The timing at which the values of the TDR0nH, TDR0nL, TDR0pH, and TDR0pL registers are loaded to the TCR0nH, TCR0nL, TCR0pH, and TCR0pL registers is upon occurrence of INTTM0n of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTM0n of the master channel, the TO0p pin cannot output the expected waveform. To rewrite both the TDR0nH and TDR0nL registers of the master and the TDR0pH and TDR0pL registers of the slave, therefore, be sure to rewrite the four registers immediately after INTTM0n is generated from the master channel.

<R> **2. To use the PWM function in 8-bit timer mode, set 00H in the higher 8 bits (TDR0nH) of the TDR0n register (master).**

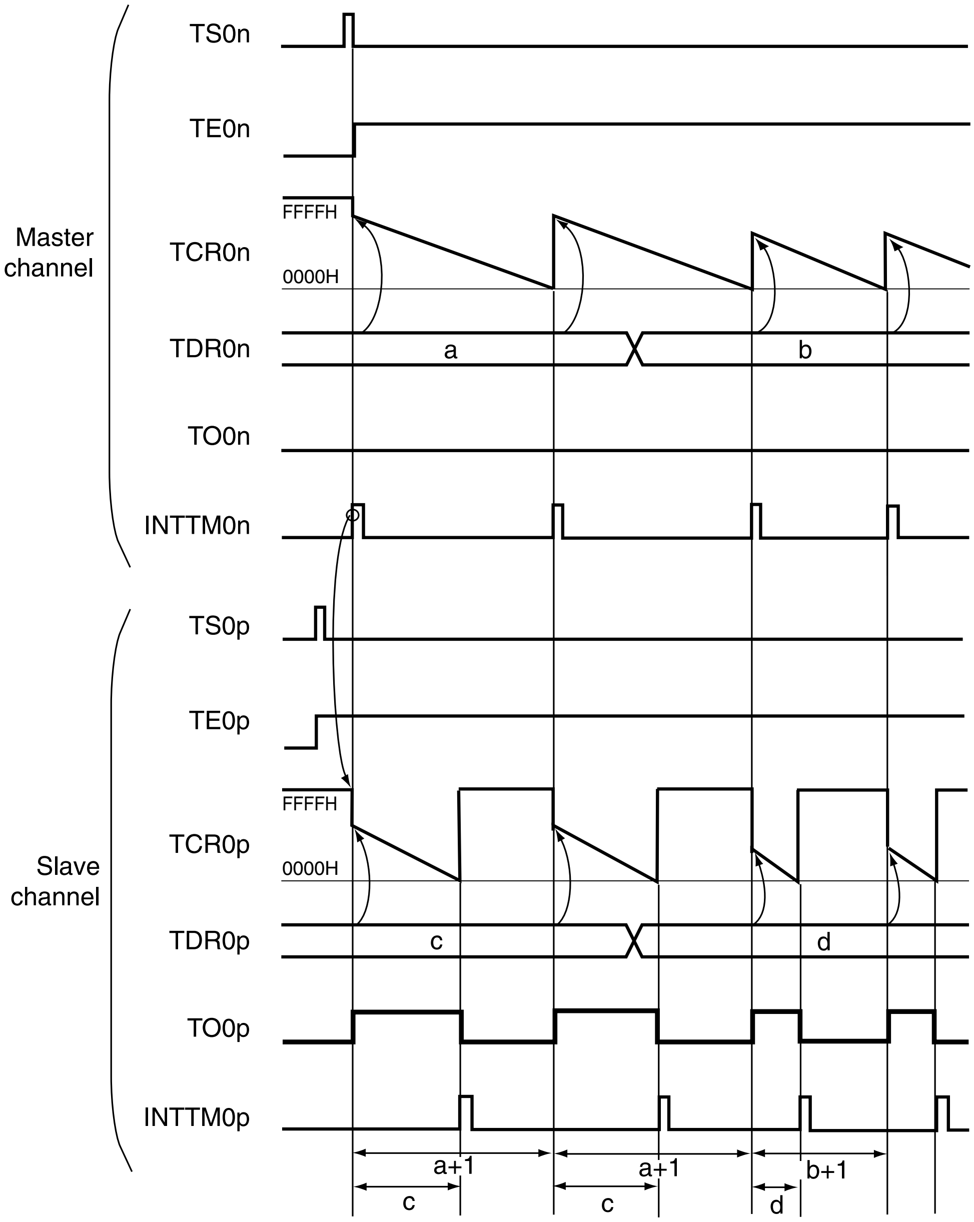
Remark n: Master channel number (n = 0)
p: Slave channel number (p = 1)

Figure 6-67. Block Diagram of Operation as PWM Function



Remark n: Master channel number (n = 0)
p: Slave channel number (p = 1)

Figure 6-68. Example of Basic Timing of Operation as PWM Function



Remark 1. n: Master channel number (n = 0)

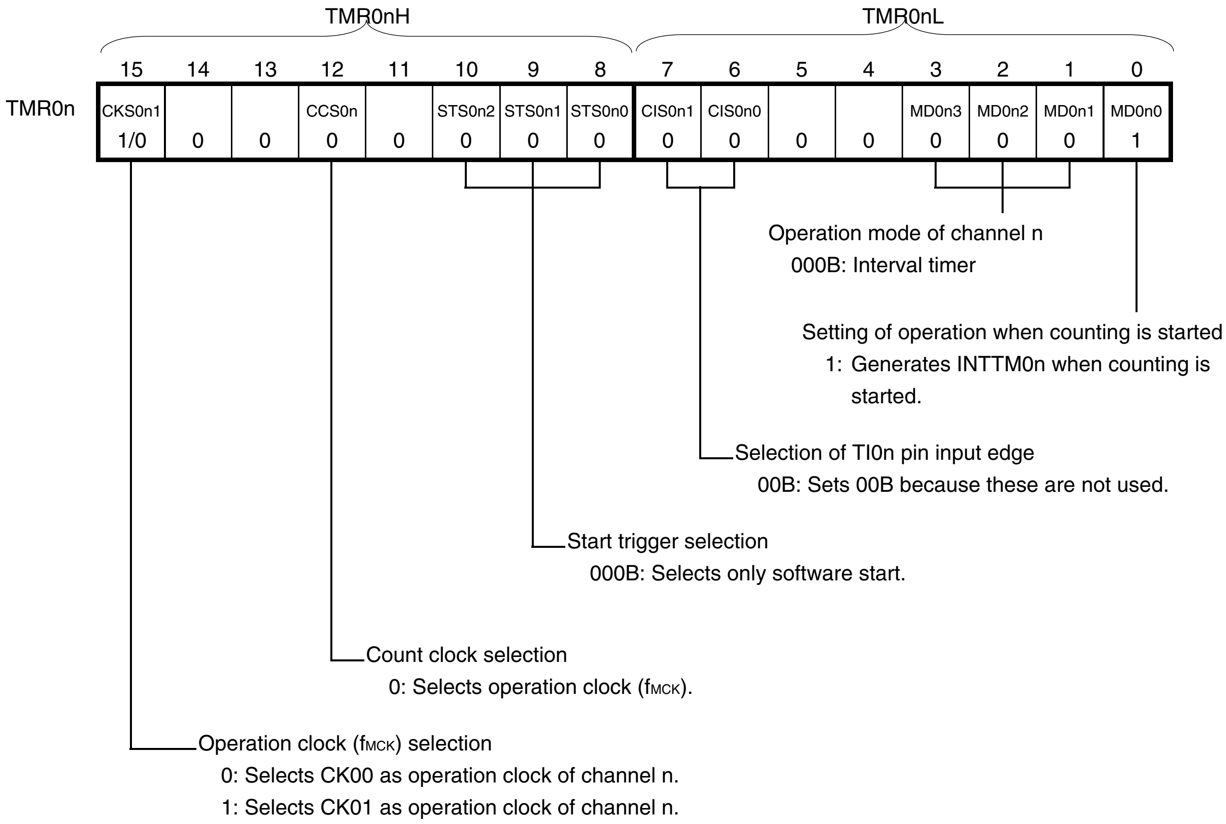
p: Slave channel number (p = 1)

2.

TS0n, TS0p:	Bit n, p of timer channel start register 0 (TS0)
TE0n, TE0p:	Bit n, p of timer channel enable status register 0 (TE0)
TCR0n, TCR0p:	Timer count registers 0n, 0p (TCR0n, TCR0p)
TDR0n, TDR0p:	Timer data registers 0n, 0p (TDR0n, TDR0p)
TO0n, TO0p:	TO0n and TO0p pins output signal

Figure 6-69. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register 0n (TMR0nH, TMR0nL)



Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTM0n when counting is started.

Selection of TI0n pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

(b) Timer output register 0 (TO0)

TO0 Bit n TO0n 0

0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

TOE0 Bit n TOE0n 0

0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

TOL0 Bit n TOL0n 0

0: Cleared to 0 when TOM0n = 0 (master channel output mode).

(e) Timer output mode register 0 (TOM0)

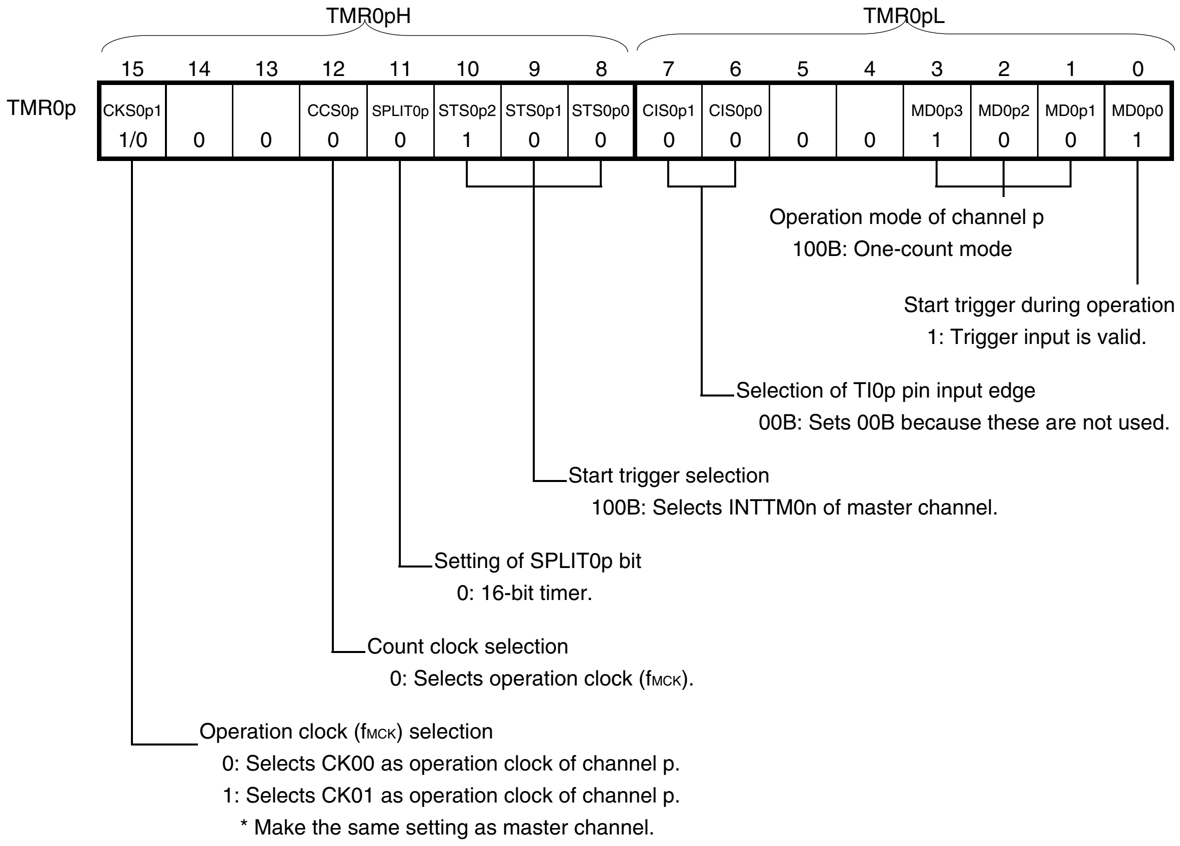
TOM0 Bit n TOM0n 0

0: Sets master channel output mode.

Remark n: Master channel number (n = 0)

Figure 6-70. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register 0p (TMR0pH, TMR0pL)



	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0p	CKS0p1 1/0	0	0	CCS0p 0	SPLIT0p 0	STS0p2 1	STS0p1 0	STS0p0 0	CIS0p1 0	CIS0p0 0	0	0	MD0p3 1	MD0p2 0	MD0p1 0	MD0p0 1

Operation mode of channel p
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TI0p pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTM0n of master channel.

Setting of SPLIT0p bit
0: 16-bit timer.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
0: Selects CK00 as operation clock of channel p.
1: Selects CK01 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register 0 (TO0)

	Bit p
TO0	TO0p 1/0

0: Outputs 0 from TO0p.
1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p
TOE0	TOE0p 1/0

0: Stops the TO0p output operation by counting operation.
1: Enables the TO0p output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit p
TOL0	TOL0p 1/0

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit p
TOM0	TOM0p 1

1: Sets the slave channel output mode.

Note		
	TMR02:	MASTER0n bit
	TMR01, TMR03:	SPLIT0p bit

Remark n: Master channel number (n = 0)
p: Slave channel number (p = 1)

Figure 6-71. 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 (when the TAU0EN bit is 0, read/write operation is disabled). →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode registers 0n, 0p (TMR0n, TMR0p) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register 0n (TDR0n) of the master channel, and a duty factor is set to the TDR0p register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0p bit of timer output mode register 0 (TOM0) is set to 1 (slave channel output mode). Sets the TOL0p bit. Sets the TO0p bit and determines default level of the TO0p output. → Sets the TOE0p bit to 1 and enables operation of TO0p. → Clears the port register and port mode register to 0. →	The TO0p pin goes into Hi-Z output state. The TO0p default setting level is output when the port mode register is in output mode and the port register is 0. TO0p does not change because channel stops operating. The TO0p pin outputs the TO0p set level.

Remark n: Master channel number (n = 0)
p: Slave channel number (p = 1)

Figure 6-71. Operation Procedure When PWM Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOE0p bit (slave) to 1 (only when operation is resumed). The TS0n (master) and TS0p (slave) bits of timer channel start register 0 (TS0) are set to 1 at the same time. The TS0n and TS0p bits automatically return to 0 because they are trigger bits.	TE0n = 1, TE0p = 1 When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMR0n and TMR0p registers, TOM0n, TOM0p, TOL0n, and TOL0p bits cannot be changed. Set values of the TDR0n and TDR0p registers can be changed after INTTM0n of the master channel is generated. The TCR0n and TCR0p registers can always be read. The TSR0n and TSR0p registers are not used.	The counter of the master channel loads the TDR0n register value to timer count register 0n (TCR0n), and counts down. When the count value reaches TCR0n = 0000H, INTTM0n output is generated. At the same time, the value of the TDR0n register is loaded to the TCR0n register, and the counter starts counting down again. At the slave channel, the value of the TDR0p register is loaded to the TCR0p register, triggered by INTTM0n of the master channel, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n (master) and TT0p (slave) bits are set to 1 at the same time. The TT0n and TT0p bits automatically return to 0 because they are trigger bits.	TE0n, TE0p = 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized but holds current status.
	The TOE0p bit of slave channel is cleared to 0 and value is set to the TO0p bit.	The TO0p pin outputs the TO0p set level.
TAU stop	To hold the TO0p pin output level Clears the TO0p bit to 0 after the value to be held is set to the port register. When holding the TO0p pin output level is not necessary Setting not required.	The TO0p 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 TO0p bit is cleared to 0 and the TO0p pin is set to port mode.)

Remark n: Master channel number (n = 0)
p: Slave channel number (p = 1)

6.9 Cautions When Using Timer Array Unit

6.9.1 Cautions when using timer output

Depending on the product, a timer output and other alternate functions may be assigned to some pins. In such case, the outputs of the other alternate functions must be set to their initial states.

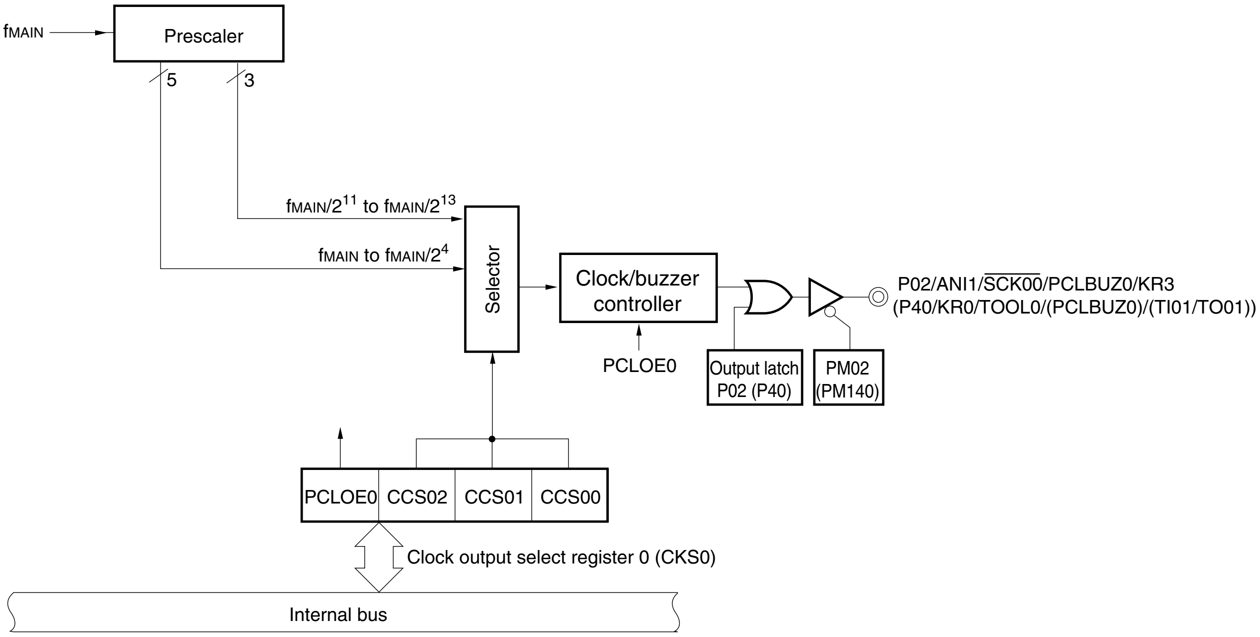
For details, see **4.5 Register Settings When an Alternate Function Is Used**.

CHAPTER 7 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

7.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for clock output for supply to peripheral ICs.
Buzzer output is a function to output a square wave of buzzer frequency.
One pin can be used to output a clock or buzzer sound.
The PCLBUZ0 pin outputs a clock selected by clock output select register 0 (CKS0).
Figure 7-1 shows the block diagram of clock output/buzzer output controller.

Figure 7-1. Block Diagram of Clock Output/Buzzer Output Controller



Caution The PCLBUZ0 pin can output a frequency, refer to 21.4 AC Characteristics.

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

7.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 7-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select register 0 (CKS0) Port mode register 0 (PM0) [Port mode register 4 (PM4)] Port register 0 (P0) [Port register 4 (P4)]

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

7.3 Registers Controlling Clock Output/Buzzer Output Controller

The following two registers are used to control the clock output/buzzer output controller.

- Clock output select register 0 (CKS0)
- Port mode register 0 (PM0) [Port mode register 4 (PM4)]

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

7.3.1 Clock output select register 0 (CKS0)

This register sets output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZ0), and sets the output clock.

The CKS0 register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-2. Format of Clock Output Select Register 0 (CKS0)

Address: FFFA5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKS0	PCLOE0	0	0	0	0	CCS02	CCS01	CCS00

PCLOE0	PCLBUZ0 pin output enable/disable specification
0	Output disable (default)
1	Output enable

CCS02	CCS01	CCS00	PCLBUZ0 pin output clock selection					
				f_{MAIN} (MHz)				
				1.25	2.5	5	10	20
0	0	0	f_{MAIN}	1.25 MHz	2.5 MHz	5 MHz$^{\text{Note}}$	10 MHz$^{\text{Note}}$	Setting prohibited$^{\text{Note}}$
0	0	1	$f_{MAIN}/2$	625 kHz	1.25 MHz	2.5 MHz	5 MHz$^{\text{Note}}$	10 MHz$^{\text{Note}}$
0	1	0	$f_{MAIN}/2^2$	312.5 kHz	625 kHz	1.25 MHz	2.5 MHz	5 MHz$^{\text{Note}}$
0	1	1	$f_{MAIN}/2^3$	156.3 kHz	312.5 kHz	625 kHz	1.25 MHz	2.5 MHz
1	0	0	$f_{MAIN}/2^4$	78.1 kHz	156.3 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	1	$f_{MAIN}/2^{11}$	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	0	$f_{MAIN}/2^{12}$	305 Hz	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	1	$f_{MAIN}/2^{13}$	153 Hz	305 Hz	610 Hz	1.22 kHz	2.44 kHz

Note The available output clock varies depending on the operating voltage range. For detail, refer to **21.4 AC Characteristics**.

<R> **Cautions 1. Change the output clock after disabling the PCLBUZ0 pin output (PCLOE0 = 0).**

2. To shift to STOP mode, execute the STOP instruction when at least 1.5 cycles of the clock used for the PCLBUZ0 pin output have elapsed after the PCLBUZ0 pin output has been disabled.

Remark f_{MAIN}: Main system clock frequency

7.3.2 Port mode registers 0, 4 (PM0, PM4)

These registers set input/output of ports 0, 4 in 1-bit units.

When using the P02/ANI1/$\overline{\text{SCK00}}$/PCLBUZ0/KR3 (P40/KR0/TOOL0/(PCLBUZ0)/(TI01/TO01)) pin for clock output and buzzer output, clear PM02 (PM40) bit and the output latch of P02 (P40) to 0. And set 0 to PMC02 bit for port mode control register 0.

The PM0 (PM4) register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 7-3. Format of Port Mode Register 0, 4 (PM0, PM4)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W

PM02	P02 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W

PM40	P40 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark The statements in parentheses are applicable when the setting of the PIOR0 bit in the PIOR register is 1.

7.4 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).

7.4.1 Operation as output pin

The PCLBUZ0 pin is output as the following procedure.

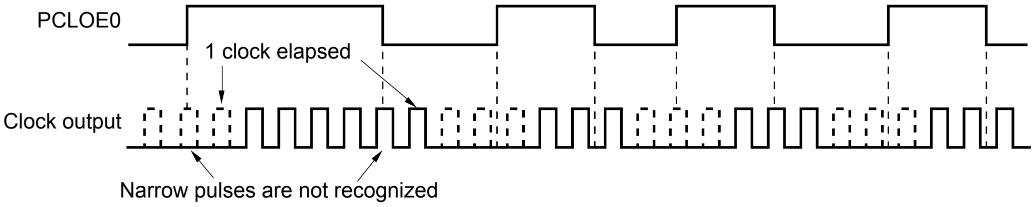
<R> <1> Set the bits in the port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx) that correspond to the pin on which the PCLBUZ0 function is multiplexed to 0.

<2> Select the output frequency with bits 0 to 2 (CCS00 to CCS02) of the clock output select register (CKS0) of the PCLBUZ0 pin (output in disabled).

<3> Set bit 7 (PCLOE0) of the CKS0 register to 1 to enable clock/buzzer output.

Remark The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOE0 bit) is switched. At this time, pulses with a narrow width are not output. Figure 7-4 shows enabling or stopping output using the PCLOE0 bit and the timing of outputting the clock.

Figure 7-4. Timing of Clock Output from PCLBUZ0



<R> **Caution Entry to STOP or HALT mode within 1.5 clock cycles of the PCLBUZ0 pin output being disabled (PCLOE0 = 0) will shorten the width of the PCLBUZ0 pin output pulse. In such cases, only execute the STOP or HALT instruction when at least 1.5 cycles of the clock used for PCLBUZ0 output have elapsed after the PCLBUZ0 pin output has been disabled.**

CHAPTER 8 WATCHDOG TIMER

8.1 Functions of Watchdog Timer

The count operation is specified by the user option byte (000C0H) in the watchdog timer.

The watchdog timer operates on the low-speed on-chip oscillator clock.

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register

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 14 RESET FUNCTION**.

When 75% of the overflow time + 3/(4 × f_{IL}) is reached, an interval interrupt can be generated.

8.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 8-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

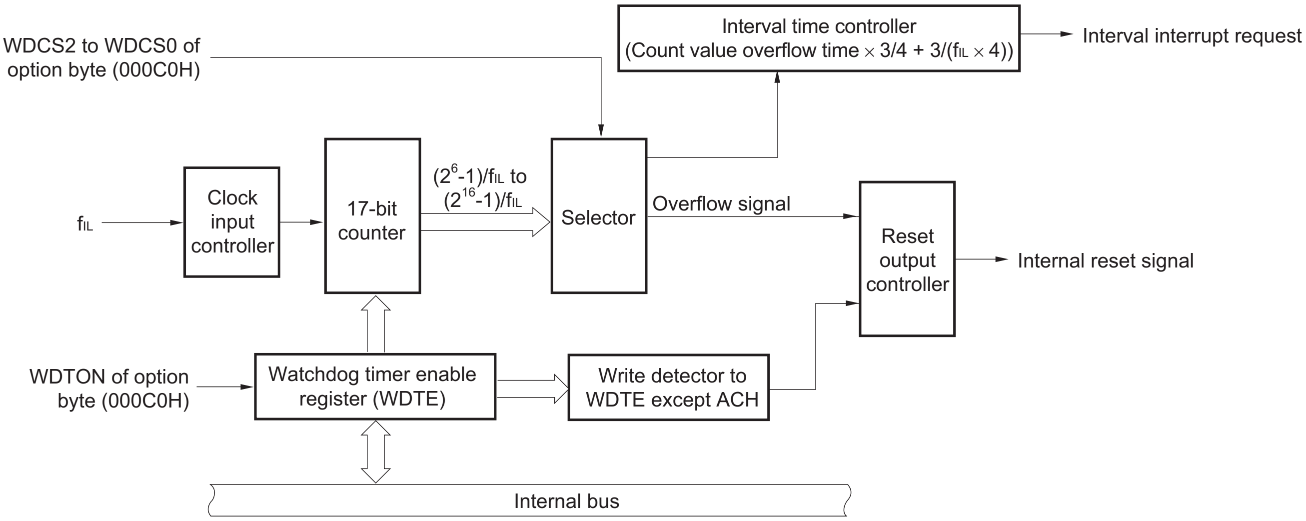
How the counter operation is controlled and overflow time are set by the option byte.

Table 8-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
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 16 OPTION BYTE**.

Figure 8-1. Block Diagram of Watchdog Timer



8.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

8.3.1 Watchdog timer enable register (WDTE)

Writing “ACH” to the WDTE register clears the watchdog timer counter and starts counting again.

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

Reset signal generation sets this register to 1AH or 9AH[Note].

Figure 8-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH After reset: 1AH/9AH[Note] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Cautions 1. If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.

2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.

3. The value read from the WDTE register is 1AH/9AH (this differs from the written value (ACH)).

8.4 Operation of Watchdog Timer

8.4.1 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 16**).

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 **8.4.2** and **CHAPTER 16**).

<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> 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. If the watchdog timer is cleared by writing “ACH” to the WDTE register, the actual overflow time may be different from the overflow time set by the option byte by up to 1/f_{IL} seconds.

2. The watchdog timer can be cleared immediately before the count value overflows.

3. The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

WDSTBYON = 0 : Watchdog timer operation stops.
WDSTBYON = 1 : Watchdog timer operation continues.

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.
If WDSTBYON = 1, setting WDTON = 0 is prohibited.

8.4.2 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) before the overflow time.

The following overflow times can be set.

When 3/4 of the overflow time + 3/(f_{IL} × 4) is reached, an interval interrupt (INTWDTI) is generated.

Table 8-3. Setting of Overflow Time and Interval Interrupt Time

WDCS2	WDCS1	WDCS0	Overflow Time (WDTRES)	Watchdog timer interval interrupt time (Count value of overflow x3/4+3/(f_{IL}x4))
0	0	0	(2^6 – 1)/f_{IL} (2.1 ms)	1.6 ms
0	0	1	(2^7 – 1)/f_{IL} (4.23 ms)	3.2 ms
0	1	0	(2^8 – 1)/f_{IL} (8.5 ms)	6.4 ms
0	1	1	(2^9 – 1)/f_{IL} (17.03 ms)	12.8 ms
1	0	0	(2^{11} – 1)/f_{IL} (68.23 ms)	51.2 ms
1	0	1	(2^{13} – 1)/f_{IL} (273.03 ms)	204.8 ms
1	1	0	(2^{14} – 1)/f_{IL} (546.1 ms)	409.6 ms
1	1	1	(2^{16} – 1)/f_{IL}(2184.5 ms)	1638.4 ms

Caution 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.

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

CHAPTER 9 A/D CONVERTER

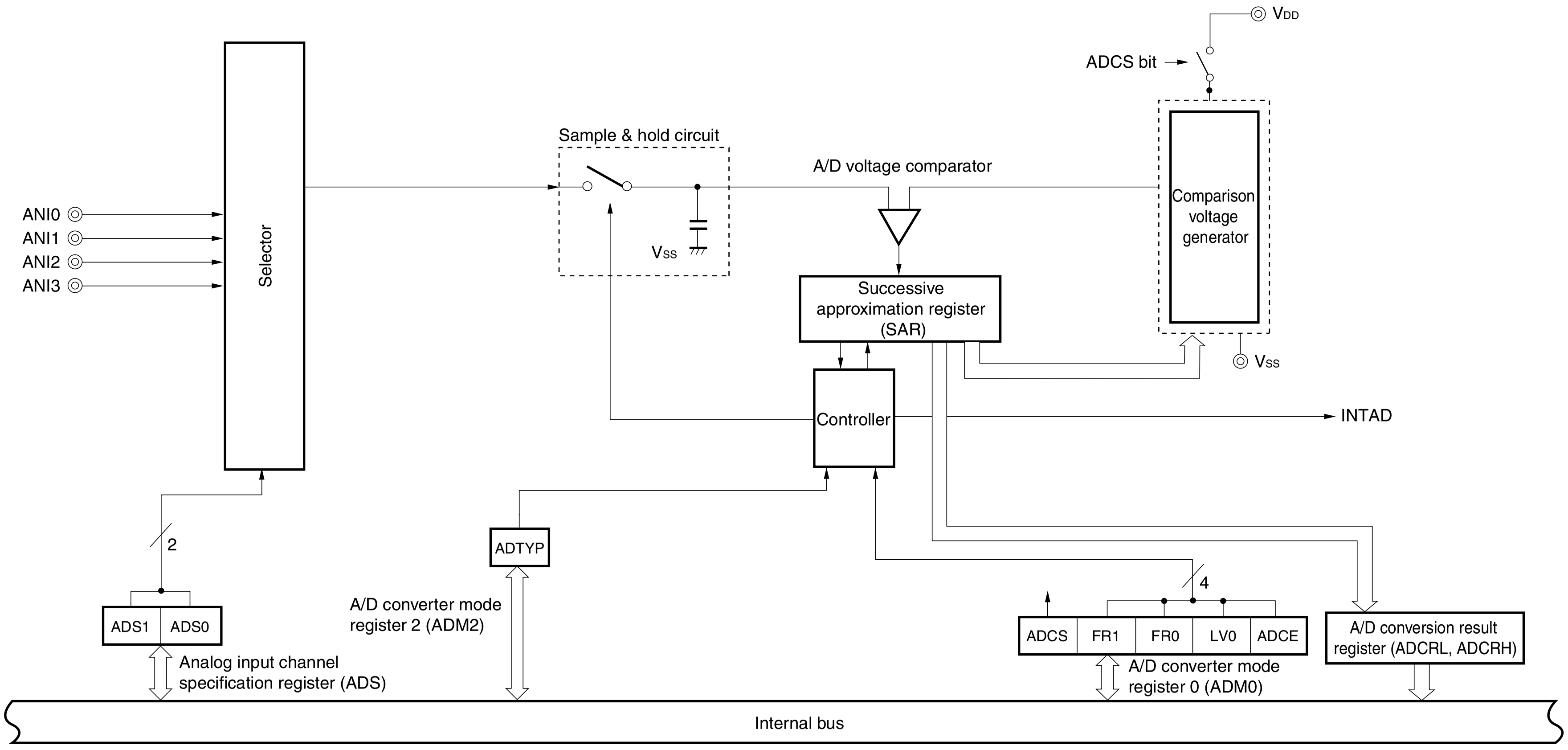
<R> 9.1 Function of A/D Converter

The A/D converter converts analog input signals into digital values, and is configured to control up to 4 channels of A/D converter analog inputs (ANI0 to ANI3). Ten-bit or eight-bit resolution can also be selected by using the ADTYP bit of A/D converter mode register 2 (ADM2).

The A/D converter has the following function.

- A/D conversion
  Software initiates the selection of one analog input channel from among ANI0 to ANI3 and the start of 10-bit or 8-bit resolution A/D conversion. An interrupt request (INTAD) is generated on completion of A/D conversion. The range of operating voltage for the A/D converter is from 2.4 to 5.5 V.

Figure 9-1. Block Diagram of A/D Converter



9.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI3 pins

These are the analog input pins of the 4 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 V_{DD}) 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 V_{DD}), 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 V_{DD})
Bit 9 = 1: (3/4 V_{DD})

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 $\geq$ Voltage tap of comparison voltage generator: Bit 8 = 1
Analog input voltage $\leq$ 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.

(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 higher bit storage register (ADCRH) and the A/D conversion result lower bit storage register (ADCRL). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(6) A/D conversion result higher bit storage register (ADCRH)

ADCRH is an 8-bit register which holds the eight higher bits of the result of 10-bit A/D conversion. The two lower bits of the result are stored in ADCRL.

(7) A/D conversion result lower bit storage register (ADCRL)

ADCRL is an 8-bit register which holds the two lower bits (ADCR1, ADCR0) of the result of 10-bit A/D conversion. The six lower bits of this register are fixed to 0.

(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.

9.3 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 2 (ADM2)
- A/D conversion result higher bit storage register (ADCRH)
- A/D conversion result lower bit storage register (ADCRL)
- Analog input channel specification register (ADS)
- Port mode control register 0 (PMC0)
- Port mode register 0 (PM0)

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 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 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	0	0	ADCEN	0	0	SAU0EN	0	TAU0EN

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

Cautions 1. When setting the A/D converter, be sure to set the following registers while the ADCEN bit is set to 1 first. 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 the port mode register 0 (PM0) and the port mode control register 0 (PMC0)).

- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 2 (ADM2)**
- **A/D conversion result higher bit storage register (ADCRH)**
- **A/D conversion result lower bit storage register (ADCRL)**
- **Analog input channel specification register (ADS)**

2. Be sure to clear the undefined bits to 0.

9.3.2 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 9-3. 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	0	FR1 Note 1	FR0 Note 1	0	LV0 Note 1	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation (conversion stopped/standby status)
1	Enables conversion operation (conversion operation status)
	<Clear conditions> • 0 is written to ADCS. • The bit is automatically cleared to 0 when A/D conversion ends. <Set condition> • 1 is written to ADCS when the ADCE bit is 1.

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 FR1, FR0, and LV0 bits and A/D conversion, see **Table 9-2 A/D Conversion Time Selection**.

2. The operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 0.1 μs from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 0.1 μs or more has elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the conversion.

<R>

Cautions 1. Only rewrite the values of the FR1, FR0, and LV0 bits in the stopped status (ADCS = 0, ADCE = 1) or in the conversion standby status (ADCS = 0, ADCE = 0). Rewriting the values of the FR1, FR0, and LV0 bits, and ADCS bits by an 8-bit manipulation instruction at the same time is prohibited.

2. Setting ADCS =1 and ADCE = 0 is prohibited.

3. Do not change the ADCE and ADCS bits from 0 to 1 at the same time by using an 8-bit manipulation instruction. Be sure to set these bits in the order described in 9.7 A/D Converter Setup Flowchart.

4. Be sure to clear the undefined bits to 0.

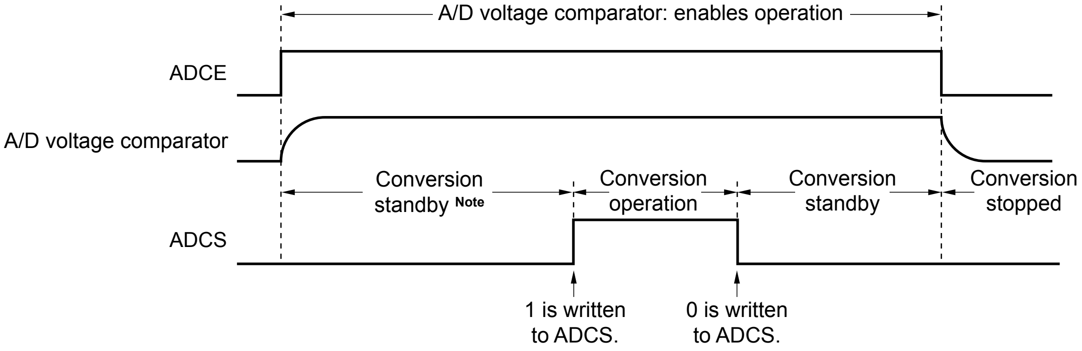
5. Setting the ADCS bit to 1 during conversion (ADCS = 1) is prohibited. When restarting the conversion for the same channel is required, stop conversion once (ADCS = 0), and then restart the A/D conversion (ADCS = 1).

When 1 is written to the ADCS bit in the conversion stopped status (ADCE = 0, ADCS = 0), the ADCS bit is not set to 1.

Table 9-1. 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

Figure 9-4. Timing Chart When A/D Voltage Comparator Is Used



Note The time from the rising of the ADCE bit to the rising of the ADCS bit must be 0.1 μs or longer to stabilize the internal circuit.

Table 9-2. A/D Conversion Time Selection

(1) 2.4 V ≤ V_{DD} ≤ 5.5 V

A/D Converter Mode Register 0 (ADM0)			Conversion Clock	Number of Conversion Clock	Conversion Time	Conversion Time Selection				
FR1	FR0	LV0				f_{CLK} = 1.25 MHz	f_{CLK} = 2.5 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	f_{CLK}/8	19 f_{AD} (Number of sampling clock: 7 f_{AD})	184/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	18.4	Setting prohibited
0	1		f_{CLK}/4		92/f_{CLK}			18.4	9.2	
1	0		f_{CLK}/2		46/f_{CLK}		18.4	9.2	4.6	
1	1		f_{CLK}		23/f_{CLK}	18.4	9.2	4.6	Setting prohibited	

(2) 2.7 V ≤ V_{DD} ≤ 5.5 V

A/D Converter Mode Register 0 (ADM0)			Conversion Clock	Number of Conversion Clock	Conversion Time	Conversion Time Selection				
FR1	FR0	LV0				f_{CLK} = 1.25 MHz	f_{CLK} = 2.5 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	f_{CLK}/8	19 f_{AD} (Number of sampling clock: 7 f_{AD})	184/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	18.4	9.2
0	1		f_{CLK}/4		92/f_{CLK}			18.4	9.2	4.6
1	0		f_{CLK}/2		46/f_{CLK}		18.4	9.2	4.6	Setting prohibited
1	1		f_{CLK}		23/f_{CLK}	18.4	9.2	4.6	Setting prohibited	
0	0	1	f_{CLK}/8	17 f_{AD} (Number of sampling clock: 5 f_{AD})	136/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	13.6	6.8
0	1		f_{CLK}/4		68/f_{CLK}			13.6	6.8	3.4
1	0		f_{CLK}/2		34/f_{CLK}		13.6	6.8	3.4	Setting prohibited
1	1		f_{CLK}		17/f_{CLK}	13.6	6.8	3.4	Setting prohibited	

Cautions 1. When rewriting the FR1, FR0, and LV0 bits to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

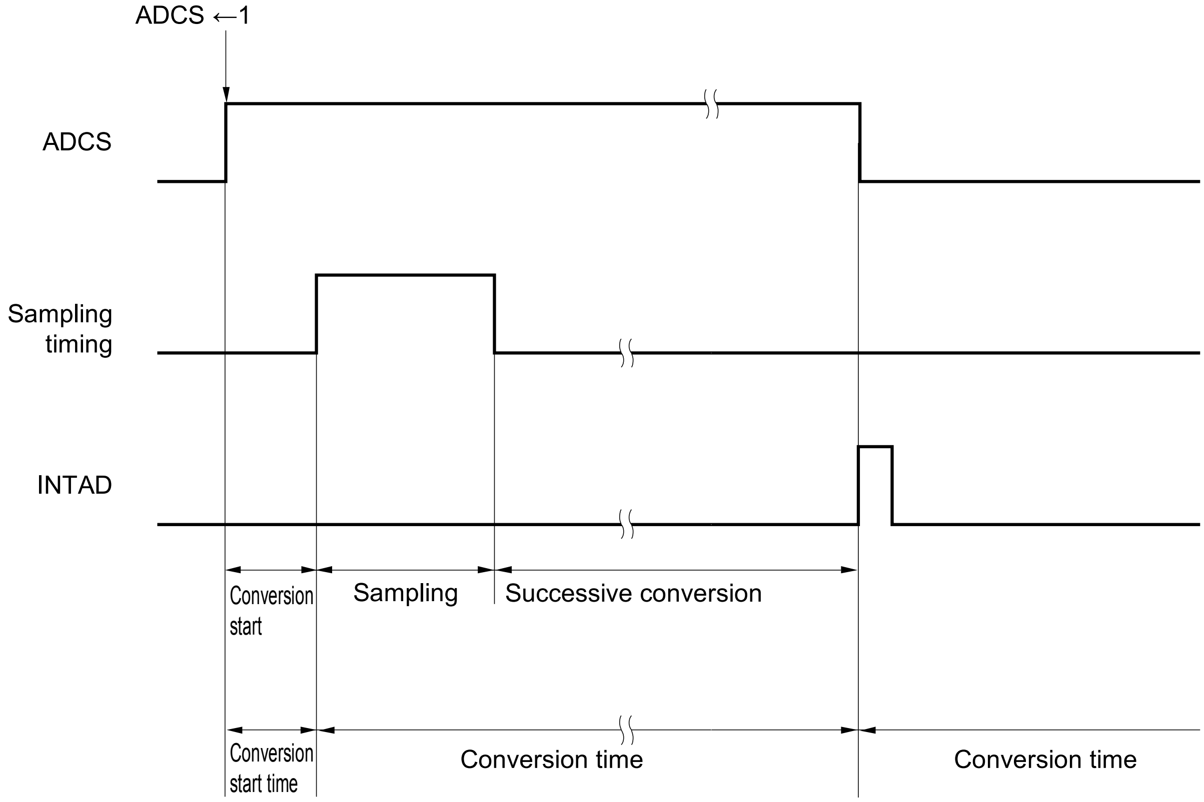
2. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

3. Do not rewrite the FR1, FR0, LV1, and ADCS bits at the same time.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 9-5. A/D Converter Sampling and A/D Conversion Timing

<R>



9.3.3 A/D converter mode register 2 (ADM2)

This register is used to select the resolution of A/D conversion.

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 9-6. Format of A/D Converter Mode Register 2 (ADM2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
ADM2	0	0	0	0	0	0	0	ADTYP

ADTYP	Resolution of A/D conversion
0	10-bit resolution
1	8-bit resolution

Caution Only rewrite the value of the ADM2 register while stopped (while the ADCS and ADCE bits of A/D converter mode register 0 (ADM0) are set to 0).

9.3.4 A/D conversion result higher bit storage register (ADCRH)

This register is an 8-bit register that holds the eight higher bits of the result of 10-bit A/D conversion. The two lower bits of the result are stored in ADCRL.

The ADCRH register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-7. Format of A/D Conversion Result Higher Bit Storage Register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH	ADCR9	ADCR8	ADCR7	ADCR6	ADCR5	ADCR4	ADCR3	ADCR2

Caution When writing to the A/D converter mode register 0 (ADM0) and the analog input channel specification register (ADS), the contents of the ADCRH/ADCRL register may become undefined. Read the conversion result following conversion completion before writing to the ADM0 and ADS registers.

9.3.5 A/D conversion result lower bit storage register (ADCRL)

This register is an 8-bit register that holds the two lower bits of the result of 10-bit A/D conversion. The six lower bits are fixed to 0.

The ADCRL register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-8. Format of A/D Conversion Result Lower Bit Storage Register (ADCRL)

Address: FFF1EH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRL	ADCR1	ADCR0	0	0	0	0	0	0

Figure 9-9 shows the state after the result of 10-bit resolution A/D conversion has been stored. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The eight higher bits of the conversion result are stored in ADCRH and the two lower bits of the result are stored in ADCRL.

Figure 9-9. The State after Storage of the Result of 10-bit Resolution A/D Conversion

ADCRH								ADCRL							
ADC R9	ADC R8	ADC R7	ADC R6	ADC R5	ADC R4	ADC R3	ADC R2	ADC R1	ADC R0	0	0	0	0	0	0

Cautions 1. When writing to the A/D converter mode register 0 (ADM0) and analog input channel specification register (ADS), the contents of the ADCRH/ADCRL registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0 and ADS registers. Using timing other than the above may cause an incorrect conversion result to be read.

2. When the ADCRL register is read while 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1), 0 is read from the higher two bits (ADCR1 and ADCR0). Note that, when the ADCRL register is read before completion of A/D conversion while 8-bit resolution A/D conversion is selected, 0 may not be read from the higher 2 bits (ADCR1, ADCR0).

9.3.6 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 9-10. 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	0	0	0	0	0	0	ADS1	ADS0

ADS1	ADS0	Analog input channel	Input source
0	0	ANI0	P01/ANI0 pin
0	1	ANI1	P02/ANI1 pin
1	0	ANI2	P03/ANI2 pin
1	1	ANI3	P04/ANI3 pin

Cautions 1. Be sure to clear the undefined bits to 0.

2. Set the port used as an analog input port to the input mode by using the port mode register 0 (PM0) and to the analog input by using the port mode control register 0 (PMC0).

<R> **3. Do not write to the ADS register during the conversion operation (ADCS = 1). Writing to the ADS register to change the analog input channel must be performed in the conversion standby status (ADCS = 0, ADCE = 1) or in the conversion stopped status (ADCS = 0, ADCE = 0)**

9.3.7 Port mode control register 0 (PMC0)

This register is used to set the digital I/O/analog input of port 0 in 1-bit units.

When using the digital I/O/analog input of port 0 as an analog input pin, set PMC01, PMC02, PMC03, and PMC04 bits to 1.

The PMC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 9-11. Formats of Port Mode Control Register 0 (PMC0)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	1	1	1	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W

PMCmn	Pmn pin digital I/O/analog input selection (m = 0; n = 1 to 4)
0	Digital I/O (dual-use function other than analog input)
1	Analog input

Cautions 1. Set the port to be set as the analog input by PMC0 and PMC1 registers to the input mode by using port mode register 0 (PM0).

2. Do not set the pin that is set by the PMC0 register as digital I/O by the analog input channel specification register (ADS).

9.3.8 Port mode register 0 (PM0)

This register is used to set the input/output of the port in 1-bit units.

When using the ANI0 to ANI3 pins as analog input ports, set PM0n bits to 1. At this time, the output latches of PM0n may be 0 or 1.

If the PM0n bits are set to 0, they cannot be used as analog input port pins.

The PM0, PMC0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Caution If a pin is set as an analog input port, not the pin level but “0” is always read.

Figure 9-12. Formats of Port Mode Register 0 (PM0)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W

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

The function of the P01/ANI0 to P04/ANI3 pins are set depending on the settings of the port mode control register 0 (PMC0), the analog input channel specification register (ADS), and the port mode register 0 (PM0).

Table 9-3. Functions of ANI0 to ANI3 Pins

PMC0	PM0	ADS	Function
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.	

9.4 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) V_{DD} 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) V_{DD}, the MSB bit of the SAR register remains set to 1. If the analog input is smaller than (1/2) V_{DD}, 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) V_{DD}
- Bit 9 = 0: (1/4) V_{DD}

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.

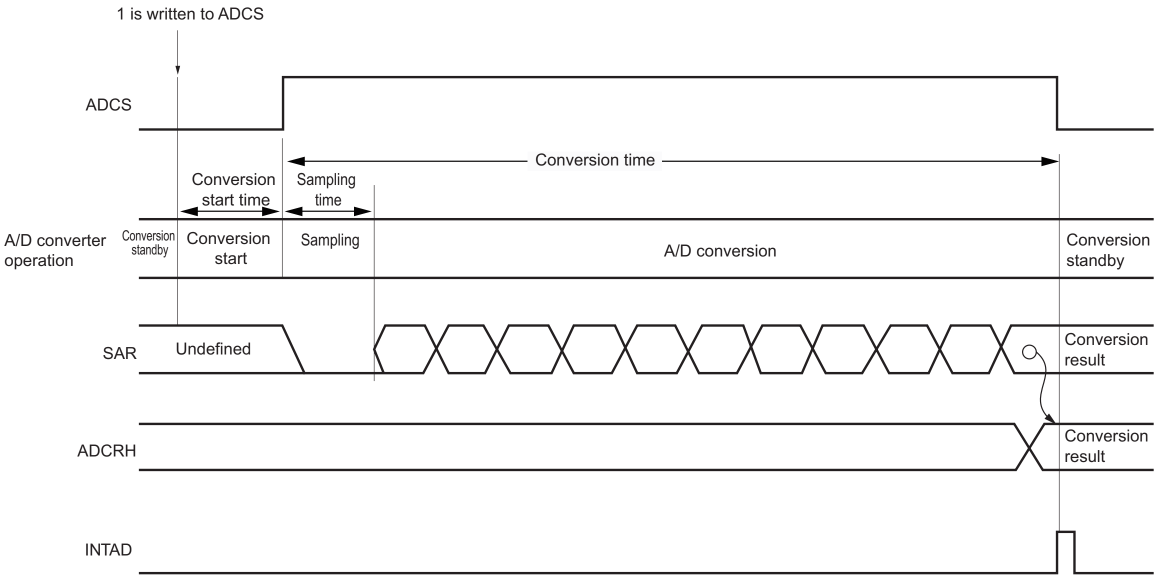
<7> Upon completion of the comparison of 10 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCRH, ADCRL) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) is generated.
After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

Remark Two types of the A/D conversion result registers are available.

• ADCRL register (8 bits):	Store the lower 2 bits of 10-bit A/D conversion value
• ADCRH register (8 bits):	Store the higher 8 bits of 10-bit A/D conversion value or 8-bit A/D conversion value

<R>

Figure 9-13. Conversion Operation of A/D Converter



A/D conversion is performed once when the bit 7 (ADCS) of the A/D converter mode register 0 (ADM0) is set to 1 by software.

Reset signal generation clears the A/D conversion result register (ADCRL, ADCRH) to 00H.

9.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI3) and the theoretical A/D conversion result (stored in the A/D conversion result register (ADCR) (= ADCRH + ADCRL)) is shown by the following expression.

$$\text{SAR} = \text{INT}\left(\frac{V_{AIN}}{V_{DD}} \times 1024 + 0.5 \times 1024 + 0.5\right)$$

$$\text{ADCR} = \text{SAR} \times 64$$

or

$$\left(\frac{\text{ADCR}}{64} - 0.5\right) \times \frac{V_{DD}}{1024} \leq V_{AIN} < \left(\frac{\text{ADCR}}{64} + 0.5\right) \times \frac{V_{DD}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

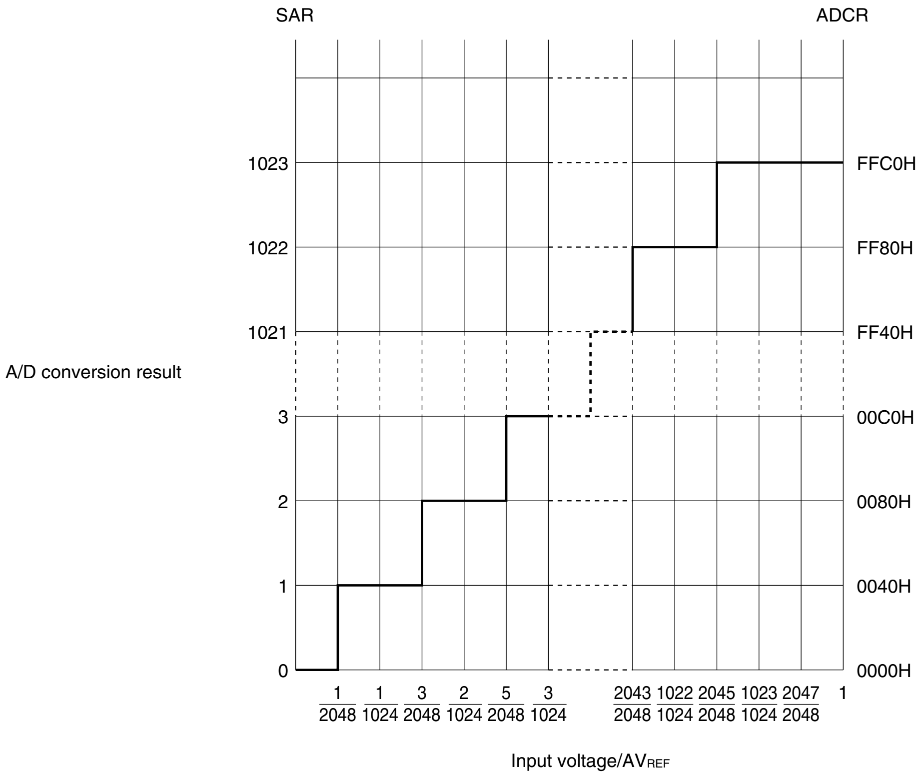
V_{AIN}: Analog input voltage

ADCR: A/D conversion result register (ADCRH + ADCRL) value

SAR: Successive approximation register

Figure 9-14 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 9-14. Relationship Between Analog Input Voltage and A/D Conversion Result



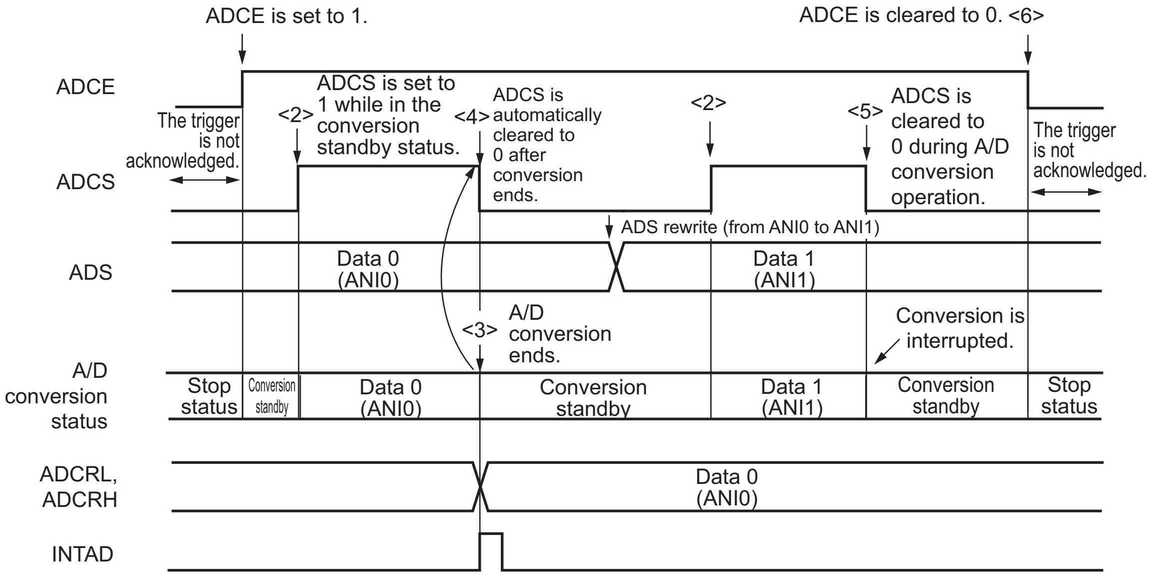
<R> 9.6 A/D Converter Operation Modes

The operation of A/D converter is described below. In addition, the procedure for specifying is described in **9.7 A/D Converter Setup Flowchart**.

<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 (0.1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).
<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCRL, 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 cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.
<6> 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.

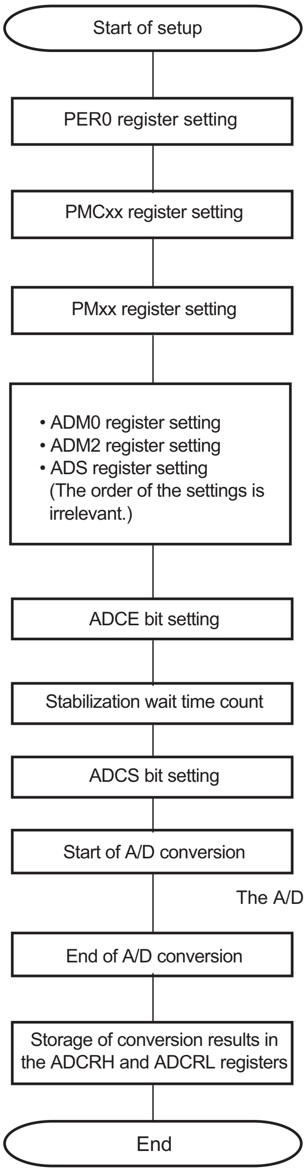
<R>

Figure 9-15. Example of Operation Timing



9.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart is described below.



The ADCEN bit of the PER0 register is set to 1, and supplying the clock starts.

The ports are set to analog input.
ANI0 to ANI6 [Note1] pins: Set using the port mode control register (PMCxx).

The ports are set to the input mode by using the port mode register (PMxx).

- A/D converter mode register 0 (ADM0)
  FR1, FR0, and LV0 bits: These are used to specify the A/D conversion time.

- A/D converter mode register 2 (ADM2)
  ADTYP bit: 8-bit/10-bit resolution

- Analog input channel specification register (ADS)
  ADS2 [Note2], ADS1, and ADS0 bits: These are used to select the analog input channels.

The ADCE bit of the ADM0 register is set to 1, and the system enters the A/D conversion standby status.

Counting 0.1   s for the stabilization wait time

After counting up to the stabilization wait time ends, the ADCS bit of the ADM0 register is set to 1, and A/D conversion starts.

The A/D conversion operations are performed.

The A/D conversion end interrupt (INTAD) is generated.

A/D conversion result higher-order bit storage register (ADCRH)
A/D conversion result lower-order bit storage register (ADCRL)
The conversion results are stored in the ADCRH and ADCRL registers.

9.8 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

9.8.1 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.

9.8.2 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.

9.8.3 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 9-16. Overall Error

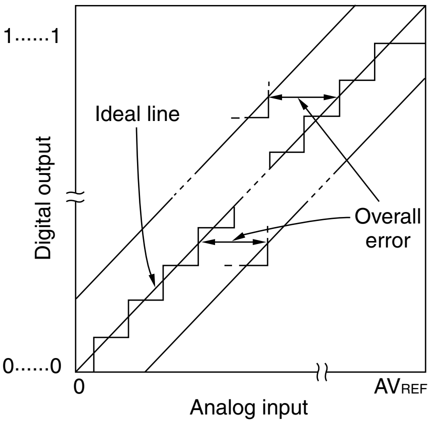
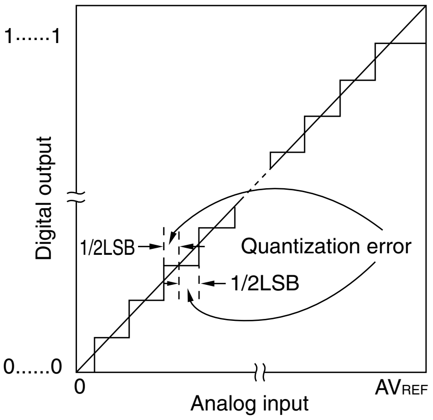


Figure 9-17. Quantization Error



9.8.4 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.

9.8.5 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.

9.8.6 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.

9.8.7 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 9-18. Zero-Scale Error

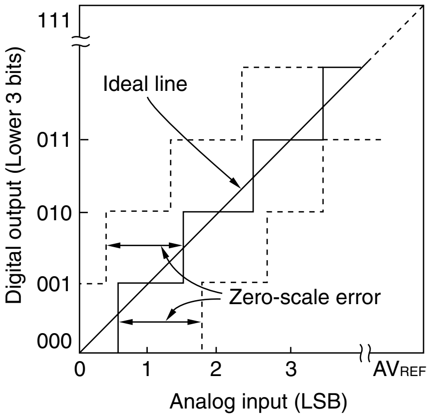


Figure 9-19. Full-Scale Error

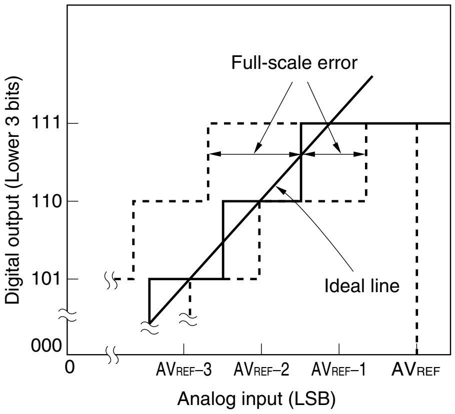


Figure 9-20. Integral Linearity Error

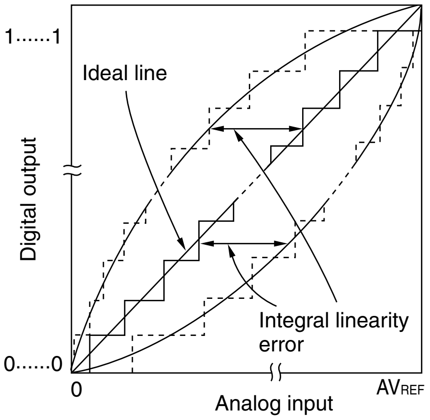
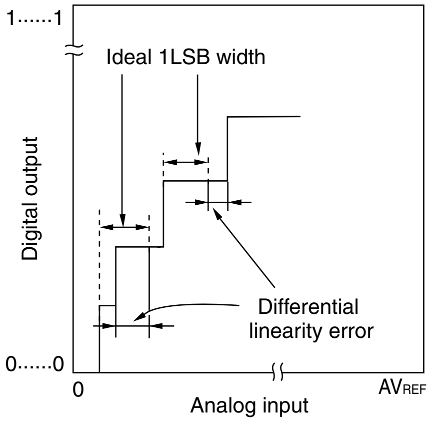


Figure 9-21. Differential Linearity Error



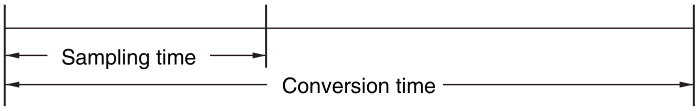
9.8.8 Conversion time

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.

9.8.9 Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



9.9 Cautions for A/D Converter

9.9.1 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.

9.9.2 Input range of ANI0 to ANI3 pins

Observe the rated range of the ANI0 to ANI3 pins input voltage. If a voltage equal to or higher than V_{DD} or equal to or lower than V_{SS} (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.

<R> 9.9.3 Conflicting operations

Writing to the ADM0 register has priority if conflict between writing to the ADCRH or ADCRL register and writing 0 to the A/D converter mode register 0 (ADM0) occurs at the end of conversion. Writing to the ADCRH or ADCRL register is not performed, nor is the conversion end interrupt signal (INTAD) generated.

9.9.4 Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the V_{DD} and ANI0 to ANI3 pins.

<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

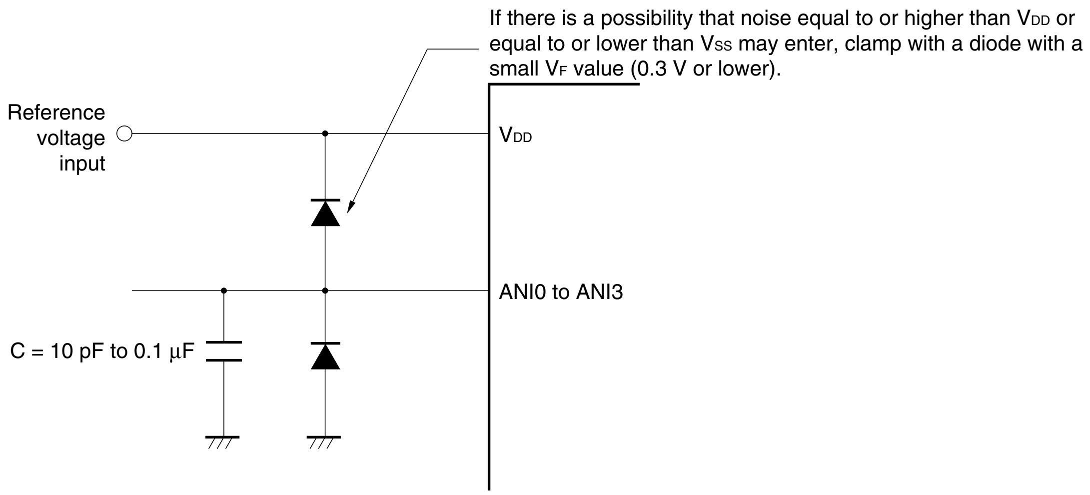
<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external C as shown in Figure 9-22 is recommended.

<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.

<R>

Figure 9-22. Analog Input Pin Connection



9.9.5 Analog input (ANIn) pins

<1> The analog input pins (ANI0 to ANI3) are also used as input port pins (P01 to P04).
When A/D conversion is performed with any of the ANI0 to ANI3 pins selected, do not change output value to alternate port P01 to P04 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to prevent such a pulse from being input or output.

9.9.6 Input impedance of analog input (ANIn) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.

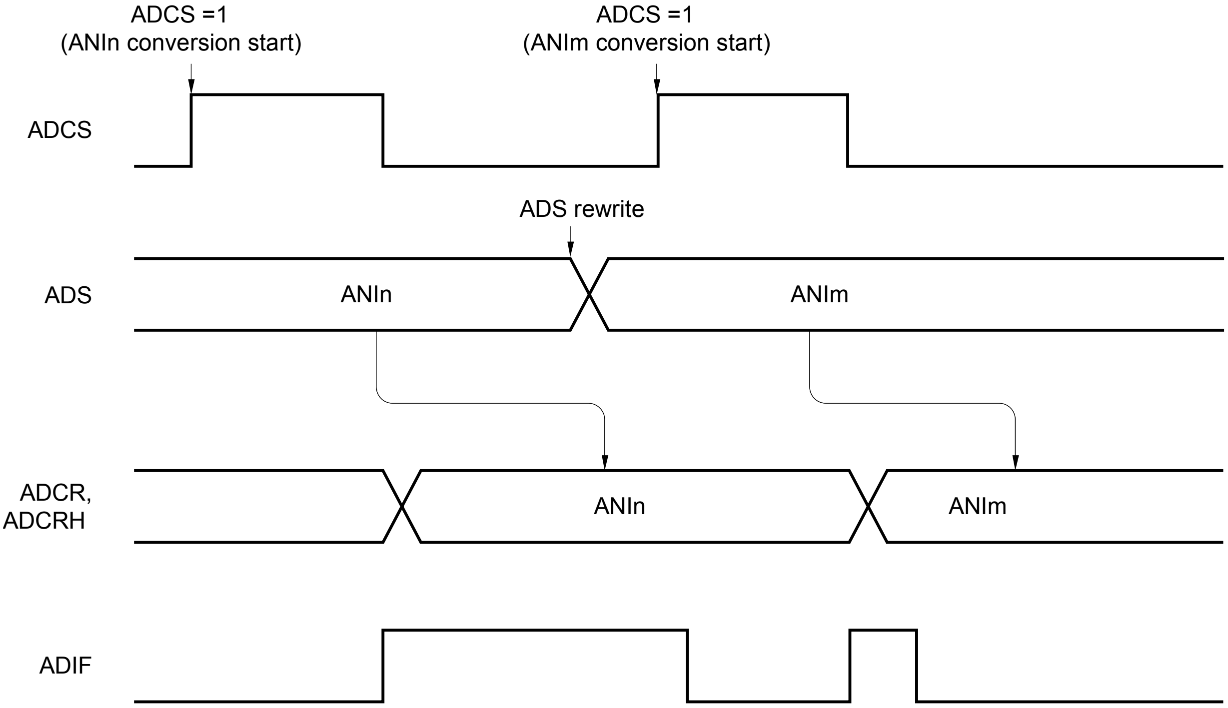
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.

To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 1 kΩ, and to connect a capacitor of about 0.1 μF to the ANI0 to ANI3 pins (see **Figure 9-22**).

9.9.7 Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed. When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

<R> **Figure 9-23. Timing of A/D Conversion End Interrupt Request Generation**



9.9.8 Conversion results just after A/D conversion start

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 0.1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

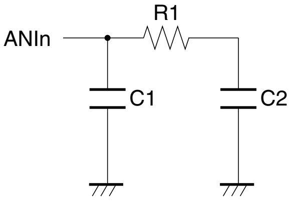
9.9.9 A/D conversion result register (ADCRH, ADCRL) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and port mode control register (PMCxx), the contents of the ADCRH and ADCRL registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, or PMCxx register.

9.9.10 Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 9-24. Internal Equivalent Circuit of ANIn Pin



<R> **Table 9-4. Resistance and Capacitance Values of Equivalent Circuit**

V_{DD}	Pins	R1 [kΩ]	C1 [pF]	C2 [pF]
2.7 V ≤ V_{DD} ≤ 5.5 V	ANI0 to ANI3	40	8	1.7
2.4 V ≤ V_{DD} ≤ 2.7 V	ANI0 to ANI3	200		1.7

Remark The resistance and capacitance values shown in Table 9-4 are not guaranteed values.

9.9.11 Starting the A/D converter

Start the A/D converter after the V_{DD} voltage stabilizes.

CHAPTER 10 SERIAL ARRAY UNIT

Serial array unit 0 has two serial channels. Each channel can achieve 3-wire serial (CSI) and UART communication. Function assignment of each channel supported by the R7F0C80112ESP, R7F0C80212ESP is as shown below.

Unit	Channel	Used as CSI	Used as UART
0	**0**	**CSI00**	**UART0**
	1	–	

A single channel cannot be used under multiple communication methods.

10.1 Functions of Serial Array Unit

Each serial interface supported by the R7F0C80112ESP, R7F0C80212ESP has the following features.

10.1.1 3-wire serial I/O (CSI00)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel.

3-wire serial 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 **10.5 Operation of 3-Wire Serial I/O (CSI00) Communication.**

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- Level setting of transmit/receive data

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate[Note]
  During master communication: Max. $f_{CLK}/4$
  During slave communication: Max. $f_{CLK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics (see **CHAPTER 21 ELECTRICAL SPECIFICATIONS**).

10.1.2 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 **10.6 Operation of UART (UART0) Communication**.

[Data transmission/reception]

- Data length of 7 or 8 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

10.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 10-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits
Buffer register	Serial data register 0n (SDR0nH, SDR0nL)
Serial clock I/O	$\overline{\text{SCK00}}$ pin (for 3-wire serial I/O)
Serial data input	SI00 pin (for 3-wire serial I/O), RxD0 pin (for UART)
Serial data output	SO00 pin (for 3-wire serial I/O), TxD0 pin (for UART), output control circuit
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register 0 (SPS0) • Serial channel enable status register 0 (SE0) • Serial channel start register 0 (SS0) • Serial channel stop register 0 (ST0) • Serial output enable register 0 (SOE0) • Serial output register 0 (SO0) • Serial clock output register 0 (CKO0) • Serial output level register 0 (SOL0) • Noise filter enable register 0 (NFEN0) • Input switch control register (ISC)
	<Registers of each channel> • Serial data register 0n (SDR0nH, SDR0nL[Note]) • Serial mode register 0n (SMR0nH, SMR0nL) • Serial communication operation setting register 0n (SCR0n) • Serial status register 0n (SSR0n) • Serial flag clear trigger register 0n (SIR0n)
	• Port output mode register 0 (POM0) • Port mode control register 0 (PMC0) • Port mode register 0 (PM0) • Port register 0 (P0)

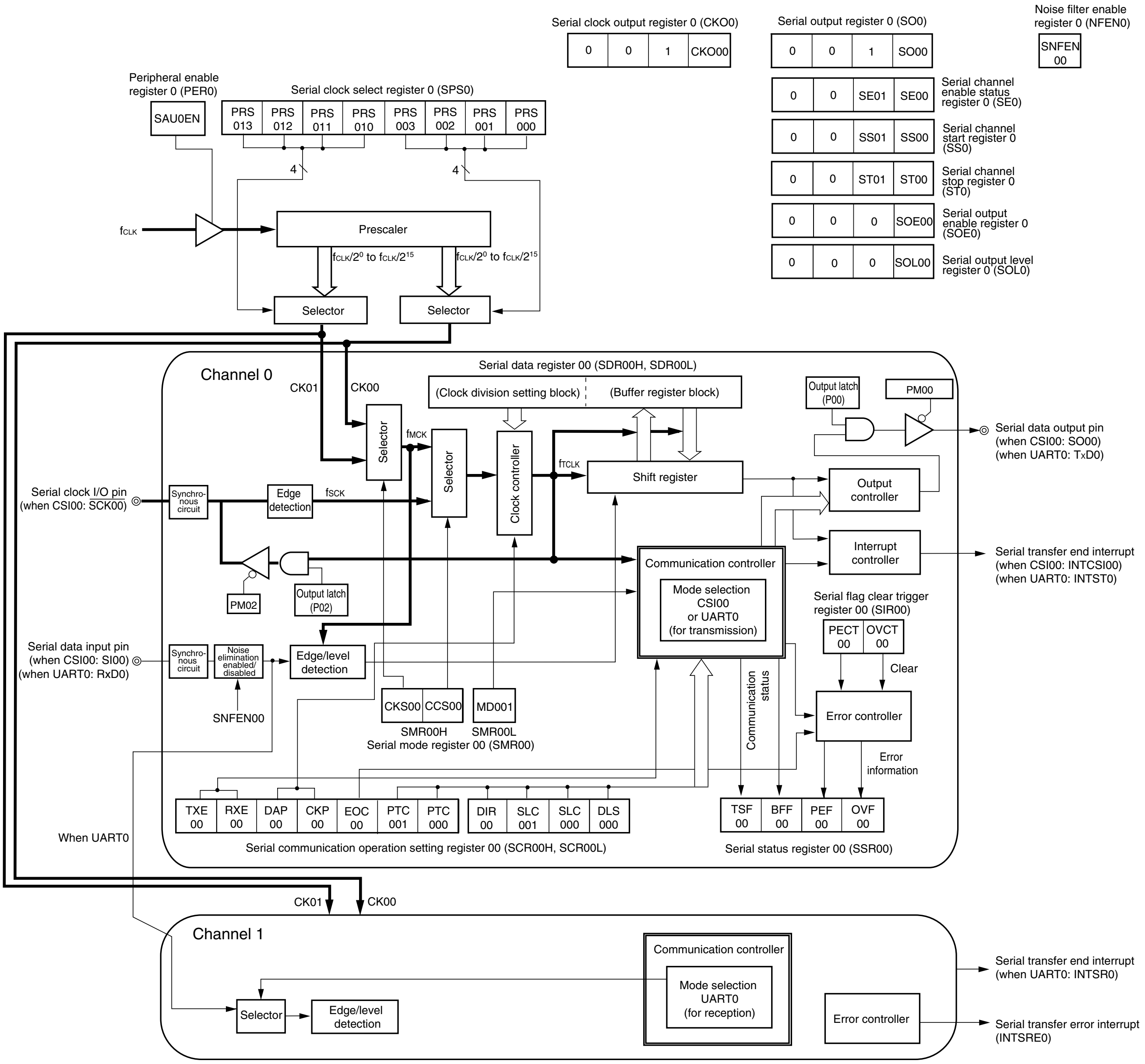
Note The serial data register 0nL (SDR0nL) can be read or written as the following SFR, depending on the communication mode.

- During CSIp communication: SIOp (CSIp data register)
- During UART0 reception: RXD0 (UARTq receive data register)
- During UART0 transmission: TXD0 (UARTq transmit data register)

Remark n: Channel number (n = 0, 1), p: CSI number (p = 00), q: UART number (q = 0)

Figure 10-1 shows the block diagram of the serial array unit 0.

Figure 10-1. Block Diagram of Serial Array Unit 0



10.2.1 Shift register

This is a 9-bit register that converts parallel data into serial data or vice versa.

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write to the shift register, use the lower 8 bits of serial data register 0nL (SDR0nL).

	8	7	6	5	4	3	2	1	0
Shift register									

10.2.2 Serial data register 0n (SDR0nH, SDR0nL)

The SDR0nH and SDR0nL registers are the transmit/receive data registers (8 bits) of channel n. SDR00H and SDR01L function as a transmit/receive buffer register, and bits 7 to 1 in the SDR00H and SDR01L registers are used as a register that sets the division ratio of the operation clock (f_{MCK}, f_{CLK}).

When data is received, parallel data converted by the shift register is stored in the SDR00H and SDR01L registers. When data is to be transmitted, set transmit data to be transferred to the shift register in the SDR00H and SDR01L registers.

The length of data stored in the SDR00H and SDR01L registers is as follows, depending on the setting of bits 0 (DLS0n0) of serial communication operation setting register 0n (SCR0nL), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of SDR0nL register)
- 8-bit data length (stored in bits 0 to 7 of SDR0nL register)

The SDR0nH and SDR0nL registers can be read or written in 8-bit units.

The SDR0nL register can be read or written in 8-bit units as the following SFR, depending on the communication mode. Note, however, writing in 8-bits units is prohibited when the operation is stopped (SE0n = 0).

- During CSIp communication: SIOp (CSIp data register)
- During UART0 reception: RXD0 (UARTq receive data register)
- During UARTq transmission: TXD0 (UARTq transmit data register)

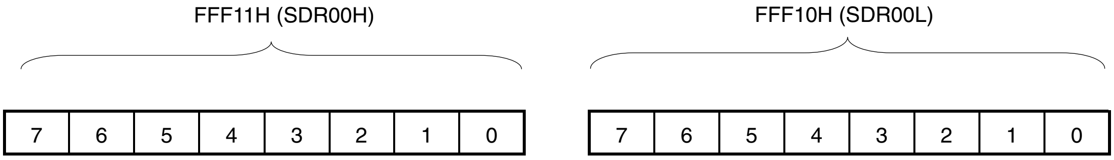
Reset signal generation clears the SDR0nH and SDR0nL registers to 00H.

Remarks 1. After completion of data reception, bits 0 to 7 which are not overwritten as a result will hold the value 0.

2. n: Channel number (n = 0, 1), p: CSI number (p = 00), q: UART number (q = 0)

Figure 10-2. Format of Serial Data Register 0n (SDR0nH, SDR0nL) (n = 0, 1)

Address: FFF10H (SDR00L), FFF11H (SDR00H), After reset: 00H R/W
FFF12H (SDR01L), FFF13H (SDR01H)



Remark For the function of the higher 7 bits of the SDR0nH register, see **10.3 Registers Controlling Serial Array Unit**.

10.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register 0 (SPS0)
- Serial mode register 0n (SMR0nH, SMR0nL)
- Serial communication operation setting register 0n (SCR0nH, SCR0nL)
- Serial data register 0n (SDR0nH, SDR0nL)
- Serial flag clear trigger register 0n (SIR0n)
- Serial status register 0n (SSR0n)
- Serial channel start register 0 (SS0)
- Serial channel stop register 0 (ST0)
- Serial channel enable status register 0 (SE0)
- Serial output enable register 0 (SOE0)
- Serial output level register 0 (SOL0)
- Serial output register 0 (SO0)
- Serial clock output register (CKO0)
- Noise filter enable register 0 (NFEN0)
- Input switch control register (ISC)
- Port output mode register 0 (POM0)
- Port mode control register 0 (PMC0)
- Port mode register 0 (PM0)
- Port register 0 (P0)

Remark n: Channel number (n = 0, 1)

10.3.1 Peripheral enable register 0 (PER0)

PER0 is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.

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

Reset signal generation clears the PER0 register to 00H.

Figure 10-3. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

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

Cautions 1. When setting serial array unit 0, be sure to set the following registers while the SAU0EN bit is set to 1 first. If SAU0EN = 0, writing to a control register of serial array unit 0 is ignored, and, even if the register is read, only the default value is read (except for the noise filter enable register 0 (NFEN0), input switch control register (ISC), port output mode register 0 (POM0), port mode register 0 (PM0), port mode control register 0 (PMC0), and port register 0 (P0)).

- **Serial clock select register 0 (SPS0)**
- **Serial mode register 0n (SMR0nH, SMR0nL)**
- **Serial communication operation setting register 0n (SCR0nH, SCR0nL)**
- **Serial data register 0n (SDR0nH, SDR0nL)**
- **Serial flag clear trigger register 0n (SIR0n)**
- **Serial status register 0n (SSR0n)**
- **Serial channel start register 0 (SS0)**
- **Serial channel stop register 0 (ST0)**
- **Serial channel enable status register 0 (SE0)**
- **Serial output enable register 0 (SOE0)**
- **Serial output level register 0 (SOL0)**
- **Serial output register 0 (SO0)**
- **Serial clock output register (CKO0)**

2. Be sure to clear the undefined bits to 0.

10.3.2 Serial clock select register 0 (SPS0)

The SPS0 register is an 8-bit register that is used to select two types of operation clocks (CK00, CK01) that are commonly supplied to each channel. CK01 is selected by bits 7 to 4 of the SPS0 register, and CK00 is selected by bits 3 to 0.

Rewriting the SPS0 register is prohibited when the register is in operation (when SE0n = 1).

The SPS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears the SPS0 register to 00H.

Figure 10-4. Format of Serial Clock Select Register 0 (SPS0)

Address: F0126H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SPS0	PRS013	PRS012	PRS011	PRS010	PRS003	PRS002	PRS001	PRS000

PRS 0n3	PRS 0n2	PRS 0n1	PRS 0n0		Section of operation clock (CK00. CK01)Note				
					f_{CLK} = 1.25 MHz	f_{CLK} = 2.5 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	1.25 MHz	2.5 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	625 kHz	1.25 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^{2}$	313 kHz	625 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^{3}$	156 kHz	313 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^{4}$	78 kHz	156 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^{5}$	39 kHz	78 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^{6}$	19.5 kHz	39 kHz	78 kHz	156 kHz	313 kHz
0	1	1	1	$f_{CLK}/2^{7}$	9.8 kHz	19.5 kHz	39 kHz	78 kHz	156 kHz
1	0	0	0	$f_{CLK}/2^{8}$	4.9 kHz	9.8 kHz	19.5 kHz	39 kHz	78 kHz
1	0	0	1	$f_{CLK}/2^{9}$	2.5 kHz	4.9 kHz	9.8 kHz	19.5 kHz	39 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.22 kHz	2.5 kHz	4.9 kHz	9.8 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	625 Hz	1.22 kHz	2.5 kHz	4.9 kHz	9.8 kHz
1	1	0	0	$f_{CLK}/2^{12}$	313 Hz	625 Hz	1.22 kHz	2.5 kHz	4.9 kHz
1	1	0	1	$f_{CLK}/2^{13}$	152 Hz	313 Hz	625 Hz	1.22 kHz	2.5 kHz
1	1	1	0	$f_{CLK}/2^{14}$	78 Hz	152 Hz	313 Hz	625 Hz	1.22 kHz
1	1	1	1	$f_{CLK}/2^{15}$	39 Hz	78 Hz	152 Hz	313 Hz	625 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 0 (ST0) = 03H) the operation of the serial array unit (SAU).

Remark f_{CLK}: CPU/peripheral hardware clock frequency

10.3.3 Serial mode register 0n (SMR0nH, SMR0nL)

The SMR0nH and SMR0nL registers are registers that set an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, an operation mode (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 SMR0nH and SMR0nL registers is prohibited when the register is in operation (when SE0n = 1). However, the MD0n0 bit can be rewritten during operation.

The SMR0nH and SMR0nL registers can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets the SMR0nH and SMR0nL registers to 00H and 20H, respectively.

Figure 10-5. Format of Serial Mode Register 0n (SMR0nH)

Address: F0111H (SMR00H), F0113H (SMR01H)

After reset: 00H R/W

Symbol: SMR0nH

7	6	5	4	3	2	1	0
CKS 0n	CCS 0n	0	0	0	0	0	STS 0n[Note]

CKS 0n	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CK00 set by the SPS0 register
1	Operation clock CK01 set by the SPS0 register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCS0n bit and the SDR0nH register, a transfer clock (f_{TCLK}) is generated.	

CCS 0n	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKS0n bit
1	Clock input f_{SCK} from the SCKp pin (slave transfer in CSI mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCS0n = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDR0nH register.	

STS 0n Note	Selection of start trigger source
0	Only software trigger is valid (selected for CSI and UART transmission). In CSI slave mode, valid edge of the SCK inpur.
1	Valid edge of the RxD0 pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SS0 register.	

Note Provided in the SMR01H register only. Set to bit to 0 in the SMR00H register.

Caution Do not change the value of the undefined bits (fixed to 0 or 1).

Remark n: Channel number (n = 0, 1)

Figure 10-6. Format of Serial Mode Register 0n (SMR0nL)

Address: F0110H (SMR00L), F0112H (SMR01L)

After reset: 20H R/W

Symbol: SMR0nL

7	6	5	4	3	2	1	0
0	SIS 0n0 Note	1	0	0	0	MD 0n1	MD 0n0

SIS 0n0 Note	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MD 0n1	Setting of operation mode of channel n
0	CSI mode
1	UART mode

MD 0n0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDR0nL register to the shift register.)
For successive transmission, the next transmit data is written by setting the MD0n0 bit to 1 when SDR0nL data has run out.	

Note Provided in the SMR01L register only. Set to bit to 0 in the SMR00L register.

Caution Do not change the value of the undefined bits (fixed to 0 or 1).

Remark n: Channel number (n = 0, 1)

10.3.4 Serial communication operation setting register 0n (SCR0nH, SCR0nL)

The SCR0nH and SCR0nL registers are communication operation setting registers 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 SCR0nH and SCR0nL registers is prohibited when the register is in operation (when SE0n = 1).

The SCR0nH and SCR0nL registers can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets the SCR0nH and SCR0nL registers to 00H and 87H, respectively.

Figure 10-7. Format of Serial Communication Operation Setting Register 0n (SCR0nH, SCR0nL) (1/2)

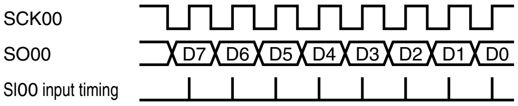
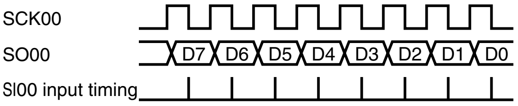
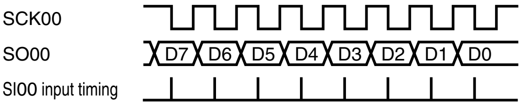
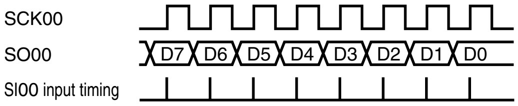
Address: F0119H (SCR00H) , F011BH (SCR01H)
After reset: 00H R/W
Symbol: SCR0nH

7	6	5	4	3	2	1	0
TXE0n	RXE0n	DAP0n	CKP0n	0	EOC0n	PTC0n1	PTC0n0

Address: F0118H (SCR00L) , F011AH (SCR01L)
After reset: 87H R/W
Symbol: SCR0nL

7	6	5	4	3	2	1	0
DIR0n	0	SLC0n1[Note 1]	SLC0n0	0	1	1	DLS0n0

TXE0n	RXE0n	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP0n	CKP0n	Selection of data and clock phase in CSI mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Be sure to set DAP0n, CKP0n = 0, 0 in the UART mode.			

EOC0n	Selection of masking of error interrupt signal (INTSREx (x = 0))
0	Masks error interrupt INTSREx (INTSRx is not masked).
1	Enables generation of error interrupt INTSREx (INTSRx is masked if an error occurs).
Set EOC0n = 0 in the CSI mode and during UART transmission[Note 2].	

Notes 1. Provided in the SCR00L register only.

2. If EOC0n is not cleared for CSI0n, error interrupt INTSREn may be generated.

Caution **Be sure to set bits 2 and 1 in the SCR0nL register to 1, and clear the undefined bits in the SCR0nH and SCR0nL registers to 0.**

Remark n: Channel number (n = 0, 1)

Figure 10-7. Format of Serial Communication Operation Setting Register 0n (SCR0nH, SCR0nL) (2/2)

Address: F0119H (SCR00H) , F011BH (SCR01H)
After reset: 00H R/W
Symbol: SCR0nH

7	6	5	4	3	2	1	0
TXE0n	RXE0n	DAP0n	CKP0n	0	EOC0n	PTC0n1	PTC0n0

Address: F0118H (SCR00L) , F011AH (SCR01L)
After reset: 87H R/W
Symbol: SCR0nL

7	6	5	4	3	2	1	0
DIR0n	0	SLC0n1[Note 1]	SLC0n0	0	1	1	DLS0n0

PTC0n1	PTC0n0	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 PTC0n1, PTC0n0 = 0, 0 in the CSI mode.			

DIR0n	Selection of data transfer sequence in CSI and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.

SLC0n1[Note 1]	SLC0n0	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 (n = 0 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 the stop bit length to 1 bit (SLC0n1, SLC0n0 = 0, 1) during UART reception. Set no stop bit (SLC0n1, SLC0n0 = 0, 0) in the CSI mode.		

DLS0n0	Setting of data length in CSI and UART modes
0	7-bit data length (stored in bits 0 to 6 of the SDR0nL register)
1	8-bit data length (stored in bits 0 to 7 of the SDR0nL register)

Notes 1. Provided in the SCR00L register only.

2. 0 is always added regardless of the data contents.

Caution Be sure to set bits 2 and 1 in the SCR0nL register to 1 and clear undefined bits in the SCR0nH and SCR0nL registers to 0.

Remark n: Channel number (n = 0, 1)

10.3.5 Serial data register 0n (SDR0nH, SDR0nL)

The SDR0nH and SDR0nL registers are the transmit/receive data registers of channel n. Considering the SDR0nH register is used as a register that sets the division ratio of the operating clock (f_{MCK}). The SDR0nL register functions as a buffer register for transmission and reception.

<R> If the CCS0n bit of the SMR0nH register is cleared to 0, the clock set by dividing the operating clock by the SDR0nH[7:1] is used as the transfer clock. If the CCS0n bit of the SMR0nH register is set to 1, SDR0nH[7:1] are fixed to 0000000B. The input clock f_{SCK} from the SCKp pin (slave transmission in the CSI mode) is used as the transfer clock.

The SDR0nL register functions as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the SDR0nL register, and during transmission, the data to be transmitted to the shift register is set to the SDR0nL register.

The SDR0nH and SDR0nL registers can be read or written in 8-bit units.

However, writing to and reading from SDR0nH[7:1] is only possible while operations are stopped (SE0n = 0). If a value is written to the SDR0nH and SDR0nL registers while the SAU is operating (SE0n = 1), the value written to the SDR0nL register is applied but writing to the SDR0nH register is ignored.

Reading from the SDR0nH register during operations (SE0n = 1) always returns the value 0.

Reset signal generation clears the SDR0nH and SDR0nL registers to 00H.

Figure 10-8. Format of Serial Data Register 0n (SDR0n)

Address: FFF10H (SDR00L) , FFF11H (SDR00H) After reset: 00H R/W
FFF12H (SDR01L) , FFF13H (SDR01H)

	FFF11H (SDR00H)								FFF10H (SDR00L)							
Symbol	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
SDR0n								0								

SDR0nH[7:1]							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	$f_{MCK}/256$

Caution Setting SDR0nH[7:1] = 0000000B to 0000001B is prohibited when UART is used. Set SDR0nH[7:1] to 0000010B or greater.

Setting SDR00H[7:1] = 0000000B is prohibited when CSI is used.

Remarks 1. For the function of the SDR0nL register, see **10.2 Configuration of Serial Array Unit**.

2. n: Channel number (n = 0, 1)

10.3.6 Serial flag clear trigger register 0n (SIR0n)

The SIR0n register is a trigger register that is used to clear each error flag of channel n.

When each bit (FECT0n, PECT0n, OVCT0n) of this register is set to 1, the corresponding bit (FEF0n, PEF0n, OVF0n) of serial status register 0n (SSR0n) is cleared to 0. Because the SIR0n register is a trigger register, it is cleared immediately when the corresponding bit of the SSR0n register is cleared.

The SIR0n register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears the SIR0n register to 00H.

Figure 10-9. Format of Serial Flag Clear Trigger Register 0n (SIR0n)

Address: F0108H (SIR00) , F010AH (SIR01) , After reset: 00H R/W

Symbol:	7	6	5	4	3	2	1	0
SIR0n	0	0	0	0	0	FECT0n^{Note}	PECT0n	OVCT0n

FECT0n^{Note}	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEF0n bit of the SSR0n register to 0.

PECT0n	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEF0n bit of the SSR0n register to 0.

OVCT0n	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVF0n bit of the SSR0n register to 0.

Note Provided in the SIR01 register only.

Caution Be sure to set undefined bits to 0

Remarks 1. n: Channel number (n = 0, 1)

2. When the SIR0n register is read, 00H is always read.

10.3.7 Serial status register 0n (SSR0n)

The SSR0n register indicates the communication status and error occurrence status of channel n. The errors indicated by this register are framing errors, parity errors, and overrun errors.

The SSR0n register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears the SSR0n register to 00H.

Figure 10-10. Format of Serial Status Register 0n (SSR0n) (1/2)

Address: F0100H (SSR00) , F0102H (SSR01) , After reset: 00H R

Symbol:	7	6	5	4	3	2	1	0
SSR0n	0	TSF0n	BFF0n	0	0	FEF0n[Note]	PEF0n	OVF0n

TSF0n	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The ST0n bit of the ST0 register is set to 1 (communication is stopped) or the SS0n bit of the SS0 register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

BFF0n	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDR0nL register.
1	Valid data is stored in the SDR0nL register.
<Clear conditions> • Transferring transmit data from the SDR0nL register to the shift register ends during transmission. • Reading receive data from the SDR0nL register ends during reception. • The ST0n bit of the ST0 register is set to 1 (communication is stopped) or the SS0n bit of the SS0 register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDR0nL registers while the TXE0n bit of the SCR0nH register is set to 1 (transmission or transmission and reception mode in each communication mode). • Receive data is stored in the SDR0nL registers while the RXE0n bit of the SCR0nH register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

Note Provided in the SSR01 register only.

Caution If data is written to the SDR0nL registers when BFF0n = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVE0n = 1) is detected.

Remark n: Channel number (n = 0, 1)

Figure 10-10. Format of Serial Status Register 0n (SSR0n) (2/2)

Address: F0100H (SSR00) - F0102H (SSR01) , After reset: 0000H R

Symbol:	7	6	5	4	3	2	1	0
SSR0n	0	TSF0n	BFF0n	0	0	FEF0n[Note]	PEF0n	OVF0n

FEF0n[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 FECT0n bit of the SIR0n register. <Set condition> • A stop bit is not detected when UART reception ends.	

PEF0n	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 PECT0n bit of the SIR0n register. <Set condition> • The parity of the transmit data and the parity bit do not match when UART reception ends (parity error).	

OVF0n	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> • 1 is written to the OVCT0n bit of the SIR0n register. <Set condition> • Even though receive data is stored in the SDR0nL registers, that data is not read and transmit data or the next receive data is written while the RXE0n bit of the SCR0nH 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 CSI mode.	

Note Provided in the SSR01 register only.

Remark n: Channel number (n = 0, 1)

10.3.8 Serial channel start register 0 (SS0)

The SS0 register is a trigger register that is used to enable communication/count for each channel.

When 1 is written to a bit of this register (SS0n), the corresponding bit (SE0n) of serial channel enable status register 0 (SE0) is set to 1 (operation is enabled). Because the SS0n bit is a trigger bit, it is cleared immediately when SE0n = 1.

The SS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears the SS0 register to 00H.

Figure 10-11. Format of Serial Channel Start Register 0 (SS0)

Address: F0122H (SS0) After reset: 00H R/W

Symbol:	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01	SS00

SS0n	Operation start trigger of channel n
0	No trigger operation
1	Sets the SE0n bit to 1 and enters the communication wait status Note.

Note If SS0n is set to 1 during transfer operations, transfer stops and the interface enters the state of waiting. At this time, the control registers and the shift register, the SCK0n and SO0n pins, and the FEF0n, PEF0n, and OVF0n flags retain their values.

Cautions 1. Be sure to clear the undefined bits to 0.

2. For the UART reception, set the RXE0n bit of SCR0nH register to 1, and then be sure to set SS0n to 1 after 4 or more f_{MCK} clocks have elapsed.

Remarks 1. n: Channel number (n = 0, 1)

2. When the SS0 register is read, 00H is always read.

10.3.9 Serial channel stop register 0 (ST0)

The ST0 register is a trigger register that is used to enable stopping communication/count for each channel.

When 1 is written to a bit of this register (ST0n), the corresponding bit (SE0n) of serial channel enable status register 0 (SE0) is cleared to 0 (operation is stopped). Because the ST0n bit is a trigger bit, it is cleared immediately when SE0n = 0.

The ST0 register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the ST0 register to 00H.

Figure 10-12. Format of Serial Channel Stop Register 0 (ST0)

Address: F0124H After reset: 00H R/W

Symbol:	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	ST01	ST00

ST0n	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SE0n bit to 0 and stops the communication operation Note

Note The control registers and the shift register, the $\overline{\text{SCK0n}}$ and SO0n pins, and the FEF0n, PEF0n, and OVF0n flags retain their values.

Caution Be sure to clear the undefined bits to 0.

Remarks **1.** n: Channel number (n = 0, 1)

2. When the ST0 register is read, 00H is always read.

10.3.10 Serial channel enable status register 0 (SE0)

The SE0 register indicates whether the data transmission/reception operation of each channel is enabled or disabled.

When 1 is written to a bit of serial channel start register 0 (SS0), the corresponding bit of this register is set to 1. When 1 is written to a bit of serial channel stop register 0 (ST0), the corresponding bit of this register is cleared to 0.

If the operation of channel n is enabled, the value of the CKO0n bit (serial clock output of channel n) of serial clock output register 0 (CKO0) cannot be rewritten by software, and a value is output from the serial clock pin according to the communication operation.

If the operation of channel n is disabled, the value of the CKO0n bit of the CKO0 register can be set by software and its value is output from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SE0 register can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the SE0 register to 00H.

Figure 10-13. Format of Serial Channel Enable Status Register 0 (SE0)

Address: F0120H (SE0) After reset: 00H R

Symbol:	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	SE01	SE00

SE0n	Indication of operation enable/disable status of channel n
0	Operation is disabled (stopped)
1	Operation is enabled.

Remark n: Channel number (n = 0, 1)

10.3.11 Serial output enable register 0 (SOE0)

The SOE0 register is used to enable or disable output of the serial communication operation of each channel.

If serial output is enabled for channel n, the value of the SO0n bit of serial output register 0 (SO0) cannot be rewritten by software, and a value is output from the serial data output pin according to the communication operation.

If serial output is disabled for channel n, the SO0n bit value of the SO0 register can be set by software, and its value is output from the serial data output pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SOE0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the SOE0 register to 00H.

Figure 10-14. Format of Serial Output Enable Register 0 (SOE0)

Address: F012AH (SOE0) After reset: 00H R/W

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	SOE00

SOE0n	Serial output enable/disable of channel n
0	Disables output by serial communication operation.
1	Enables output by serial communication operation.

Caution Be sure to clear the undefined bits to 0.

Remark n: Channel number (n = 0)

10.3.12 Serial output register 0 (SO0)

The SO0 register is a buffer register for serial output of each channel.

The value of the SO0n bit of this register is output from the serial data output pin of channel n.

The SO0n bit of this register can be rewritten by software only when serial output is disabled (SOE0n = 0). When serial output is enabled (SOE0n = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

To use the pin for serial interface as a port function pin, set the corresponding SO0n bit to 1.

The SO0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears the SO0 register to 03H.

Figure 10-15. Format of Serial Output Register 0 (SO0)

Address: F0128H (SO0) After reset: 03H R/W

Symbol:	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	1	SO00

SO0n	Serial data output of channel n
0	Serial data output value is “0”.
1	Serial data output value is “1”.

Remark n: Channel number (n = 0)

10.3.13 Serial clock output register (CKO0)

The CKO0 register is a buffer register for serial clock output of each channel.

The value of the CKO0n bit of this register is output from the serial clock output pin of channel n.

The CKO0n bit of this register can be rewritten by software only when channel operation is disabled (SE0n = 0). When channel operation is enabled (SE0n = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

To use the pin for serial interface as a port function pin, set the corresponding CKO0n bit to 1.

The CKO0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears the CKO0 register to 03H.

Figure 10-16. Format of Serial Clock Output Register (CKO0)

Address: F0129H (CKO0) After reset: 03H R/W

Symbol:	7	6	5	4	3	2	1	0
CKO0	0	0	0	0	0	0	1	CKO00

CKO0n	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

Remark n: Channel number (n = 0)

10.3.14 Serial output level register 0 (SOL0)

The SOL0 register 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 to corresponding bit in the CSI mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOE0n = 1). When serial output is disabled (SOE0n = 0), the value of the SO0n bit is output as is.

Rewriting the SOL0 register is prohibited during operation (SE0n = 1).

The SOL0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears the SOL0 register to 00H.

Figure 10-17. Format of Serial Output Level Register 0 (SOL0)

Address: F0134H (SOL0) After reset: 00H R/W

Symbol:	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	SOL00

SOL0n	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 the undefined bits to 0.

Remark n: Channel number (n = 0)

10.3.15 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 CSI communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected.

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for the target channel.

The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the NFEN0 register to 00H.

Figure 10-18. 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 (RxD0/P01)
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 other than the RxD0 pin.	

Caution Be sure to clear undefined bits to 0.

10.3.16 Input switch control register (ISC)

The ISC1 and ISC0 bits in the ISC register are used to handle the combination of the external interrupt and the timer array unit at the time of baud rate correction for reception by UART0.

When bit 0 is set to 1, the input signal on the serial data input (RxD0) pin is selected as the external interrupt input (INTP0), making detection of the wake-up signal in the form of the INTP0 interrupt possible.

When bit 1 is set to 1, the input signal on the serial data input (RxD0) pin is selected as the timer input, making measuring the pulse-width for baud rate correction possible once the wake-up signal is detected.

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

Reset signal generation clears the ISC register to 00H.

Figure 10-19. Format of Input Switch Control Register (ISC)

Address: F00730H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Switching the channel 1 of the timer array unit
0	Select the input signal on TI01 pin as the timer input (normal operation)
1	Select the input signal on RxD0 pin as the timer input (Detection of the wake-up signal and pulse-width-measurement for baud rate correction)

ISC0	Switching the external interrupt (INTP0)
0	Select the input signal on INTP0 pin as the external interrupt input (normal operation)
1	Select the input signal on RxD0 pin as the external interrupt input (detection of the wake-up signal)

Caution Be sure to clear undefined bits to 0.

10.3.17 Port output mode register 0 (POM0)

This register sets the output mode of port 1 in 1-bit units.

In addition, POM0 register is set with PMxx, PMCxx, and PUxx registers, whether or not to use the on-chip pull-up resistor (see **4.3.3 Pull-up resistor option register s 0, 4, 12 (PU0, PU4, PU12)**).

The POM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the POM0 register to 00H.

Figure 10-20. Format of Port Output Mode Register 0 (POM0)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	0	0	0	0	POM00	F0050H	00H	R/W

POM0n	P0n pin output buffer selection (n = 0)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

10.3.18 Port mode register 0 (PM0)

This register sets input/output of port 0 in 1-bit units.

When using the ports (such as P02/ANI1/$\overline{\text{SCK00}}$/PCLBUZ0/KR3) to be shared with the serial data output pin or serial clock output pin for serial data output or serial clock output, set the port mode register (PM0) and port mode control register (PMC0) bit corresponding to each port to 0. And set the port register (P0) bit corresponding to each port to 1

Example: Using P02/ANI1/$\overline{\text{SCK00}}$/PCLBUZ0/KR3 for serial clock output
Set the PMC02 bit of the port mode control register 0 to 0.
Set the PM02 bit of the port mode register 0 to 0.
Set the P02 bit of the port register 0 to 1.

When using the ports (such as P02/ANI1/$\overline{\text{SCK00}}$/PCLBUZ0/KR3) to be shared with the serial data input pin or serial clock input pin for serial data input or serial clock input, set the port mode register (PM0) bit corresponding to each port to 1. Also set the port mode control register (PMC0) bit corresponding to each port to 0. At this time, the port register (P0) bit may be 0 or 1.

Example: Using P02/ANI1/$\overline{\text{SCK00}}$/PCLBUZ0/KR3 for serial data input or serial clock input
Set the PMC02 bit of the port mode control register 0 to 0.
Set the PM02 bit of port mode register 0 to 1.
Set the P02 bit of port register 1 to 0 or 1.

The PM0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets the PM0 register to FFH.

Figure 10-21. Format of Port Mode Register 0 (PM0)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W

PM0n	Selection of P0n pin I/O mode (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

10.4 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 serial interface function alternate pins can be used as port function pins in this mode.

10.4.1 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.

Figure 10-22. Peripheral Enable Register 0 (PER0) Setting When Stopping Operation by Units

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAU0 to be stopped to 0.

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

Control of SAU0 input clock
0: Stops supply of input clock
1: Supplies input clock

Cautions 1. If SAU0EN = 0, writing to a control register of serial array unit 0 is ignored, and, even if the register is read, only the default value is read.

Note that this does not apply to the following registers.

<R>
- **Input switch control register (ISC)**
- **Noise filter enable register 0 (NFEN0)**
- **Port output mode register 0 (POM0)**
- **Port mode register 0 (PM0)**
- **Port register 0 (P0)**

2. Be sure to clear the undefined bits to 0.

Remark ▨: Setting disabled (fixed by hardware)

×: 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.

10.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 10-23. Each Register Setting When Stopping Operation by Channels

(a) Serial channel stop register 0 (ST0) … This register is a trigger register that is used to enable stopping communication/count by each channel.

Symbol:	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	ST01 0/1	ST00 0/1

1: Clears the SE0n bit to 0 and stops the communication operation

* Because the ST0n bit is a trigger bit, it is cleared immediately when SE0n = 0.

(b) Serial Channel Enable Status Register 0 (SE0) … This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

Symbol:	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	SE01 0/1	SE00 0/1

0: Operation stops

* The SE0 register is a read-only status register. Operation is stopped by using the ST0 register.
For a channel whose operation is disabled, the value of the CKO0n bit of the SO0 register can be set by software.

(c) Serial output enable register 0 (SOE0) … This register is a register that is used to enable or stop output of the serial communication operation of each channel.

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	SOE00 0/1

0: Stops output by serial communication operation

* For channel n whose serial output is stopped, the SO0n bit value of the SO0 register can be set by software.

(d) Serial clock output register 0 (CKO0) …This register is a buffer register for serial output of each channel.

Symbol:	7	6	5	4	3	2	1	0
CKO0	0	0	0	0	0	0	1	CKO00 0/1

1: Serial clock output value is “1”

* When using pins corresponding to each channel as port function pins, set the corresponding CKO0n bit to “1”.

(e) Serial output register 0 (SO0) …This register is a register that is used to set a value of serial output of each channel.

Symbol:	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	1	SO00 0/1

1: Serial data output value is “1”

* When using pins corresponding to each channel as port function pins, set the corresponding SO0n bit to “1”.

Remarks 1. n: Channel number (n = 0, 1)

2. ▨ : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

10.5 Operation of 3-Wire Serial I/O (CSI00) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- Level setting of transmit/receive data

[Clock control]

- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate

  During slave communication: Max. $f_{CLK}/6$Note

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics (see **CHAPTER 21 ELECTRICAL SPECIFICATIONS**).

Unit	Channel	Used as CSI	Used as UART
0	0	CSI00	UART0
	1	–	

3-wire serial I/O (CSI00) performs the following seven types of communication operations.

- Master transmission (See **10.5.1**.)
- Master reception (See **10.5.2**.)
- Master transmission/reception (See **10.5.3**.)
- Slave transmission (See **10.5.4**.)
- Slave reception (See **10.5.5**.)
- Slave transmission/reception (See **10.5.6**.)

10.5.1 Master transmission

Master transmission is that the R7F0C80112ESP, R7F0C80212ESP output a transfer clock and transmits data to another device.

3-Wire Serial I/O	CSI00
Target channel	Channel 0 of SAU0
Pins used	$\overline{\text{SCK00}}$, SO00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	None
Transfer data length	7 or 8 bits
Transfer rate	Max. f_{CLK}/4 [Hz] (SDR0nH[7:1] = 1 or more) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]$^{\text{Note}}$
Data phase	Selectable by the DAP0n bit of the SCR0nH register • DAP0n = 0: Data output starts at the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKP0n bit of the SCR0nH register • CKP0n = 0: Non-reverse (data output at the falling edge and data input at the rising edge of SCK) • CKP0n = 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 function characteristics in the electrical specifications (see **CHAPTER 21 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{CLK}: System clock frequency

2. n = 0

(1) Register setting

Figure 10-24. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol: SMR0nH

7	6	5	4	3	2	1	0
CKS0n	CCS0n						STS0n
0/1	0	0	0	0	0	0	0

Symbol: SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol: SCR0nH

7	6	5	4	3	2	1	0
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
1	0	0/1	0/1	0	0	0	0

Symbol: SCR0nL

7	6	5	4	3	2	1	0
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **10.3 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 0n (SDR0nH, SDR0nH)

Symbol: SDR0nH

7	6	5	4	3	2	1	0
Baud rate setting (Operation clock (f_{MCK}) division setting)							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Transmit data (Transmit data setting)							

SIOp

(d) Serial clock output register 0 (CKO0) … Sets only the bits of the target channel.

Symbol: CKO0

7	6	5	4	3	2	1	0
							CKO00
0	0	0	0	0	0	1	0/1

Communication starts when these bits are 1 if the clock phase is non-inversion (the CKP0n bit of the SCR0n = 0). If the clock phase is inverted (CKP0n = 1), communication starts when these bits are 0.

Figure 10-24. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00) (2/2)

(e) Serial output register 0 (SO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	1	SO00 0/1

(f) Serial output enable register 0 (SOE0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	SOE00 0/1

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0	SS00 0/1

Remarks 1. n = 0, p: CSI number (p = 00)

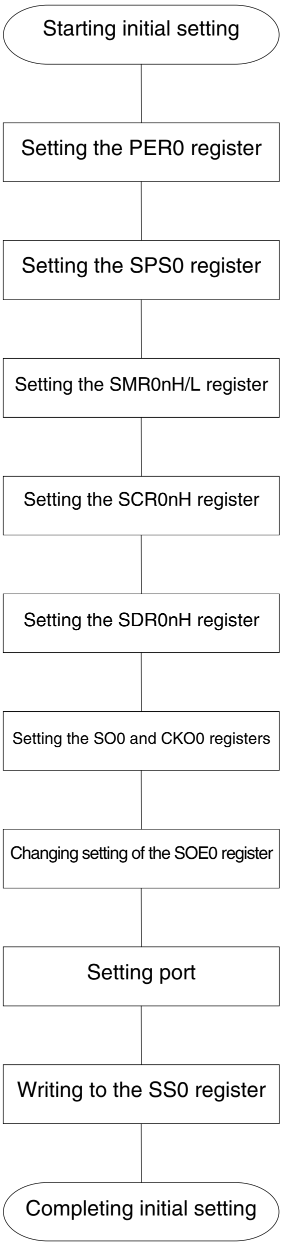
2. ☐: Setting is fixed in the CSI master transmission mode, ▨: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user.

(2) Operation procedure

Figure 10-25. 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 (CKO0n) and serial data (SO0n).

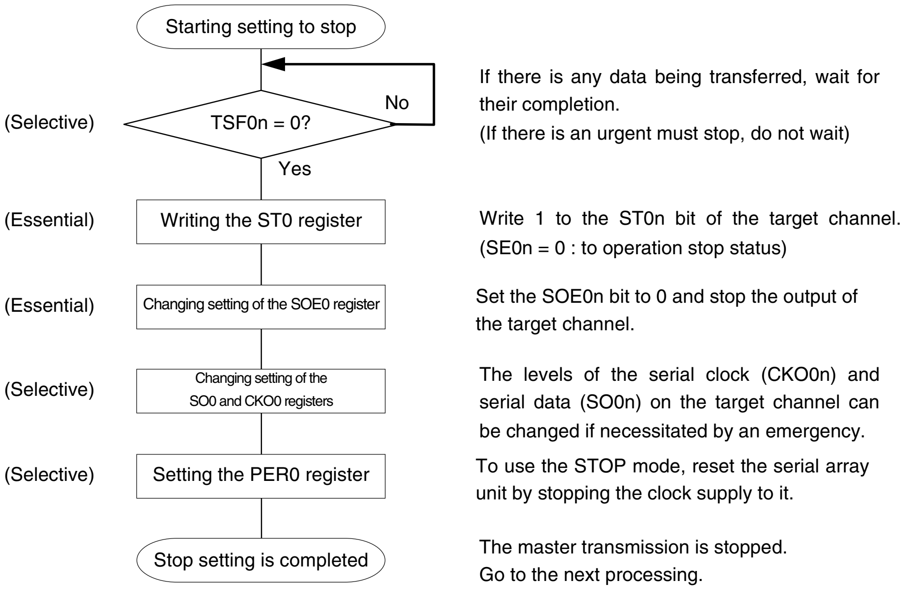
Set the SOE0n bit to 1 and enable data output of the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SS0n bit of the target channel to 1 (SE0n bit = 1 : to enable operation).

Setting of SAU is completed.
Set transmit data to the SIOp register (the SDR0nL register) and start communication.

Figure 10-26. 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 ST0n bit of the target channel.
(SE0n = 0 : to operation stop status)

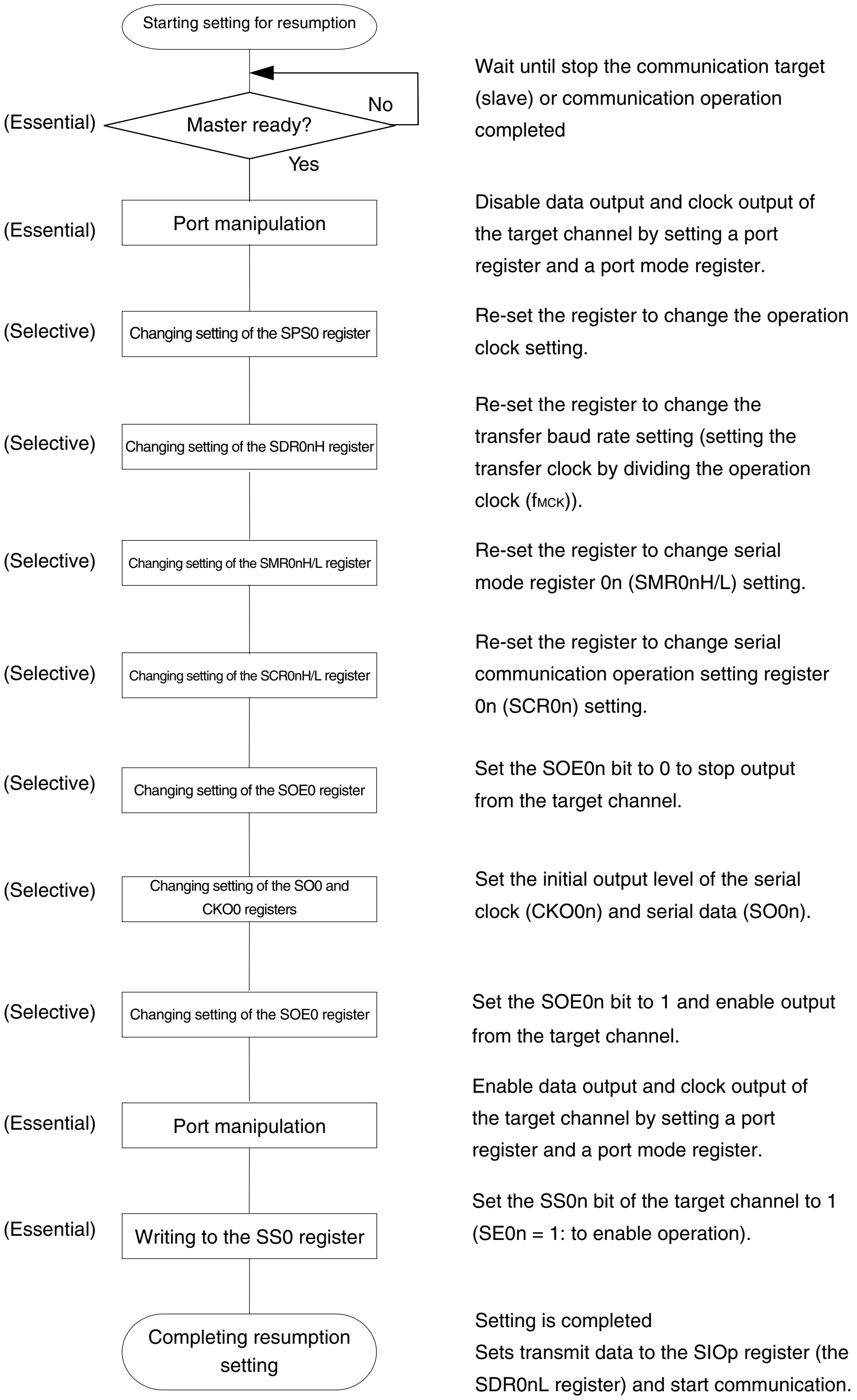
Set the SOE0n bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKO0n) and serial data (SO0n) 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.

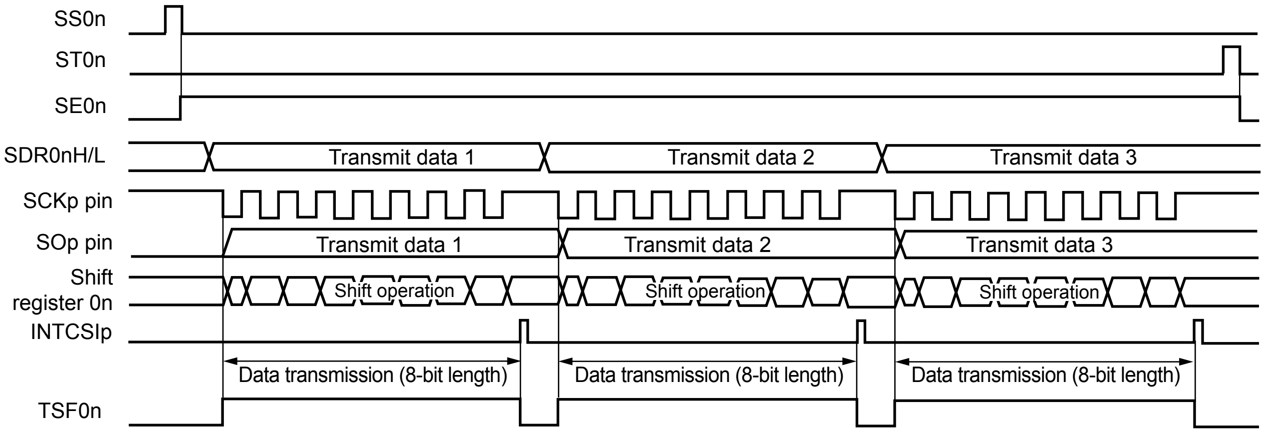
Figure 10-27. Procedure for Resuming Master Transmission



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

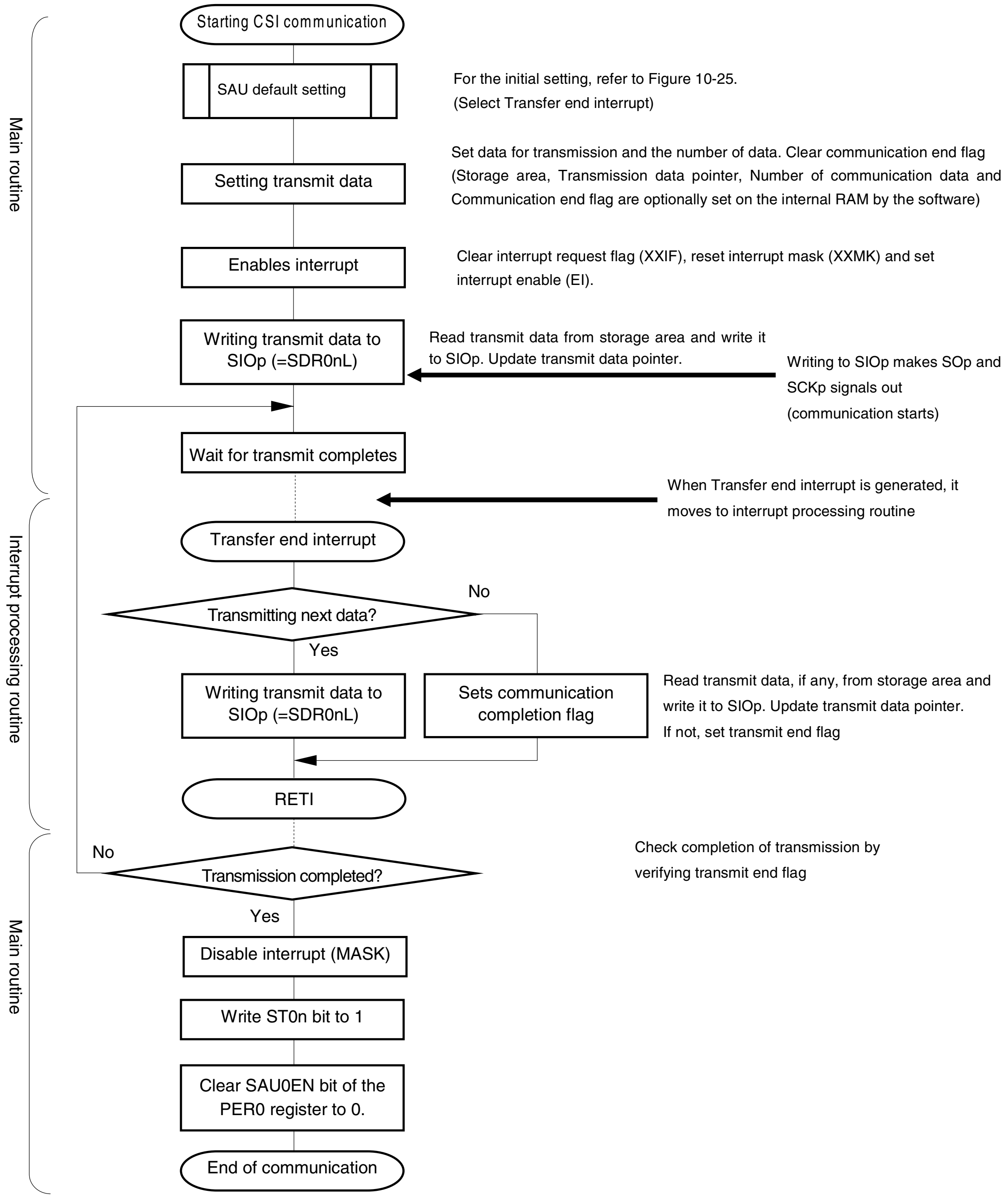
(3) Processing flow (in single-transmission mode)

Figure 10-28. Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAP0n = 0, CKP0n = 0)



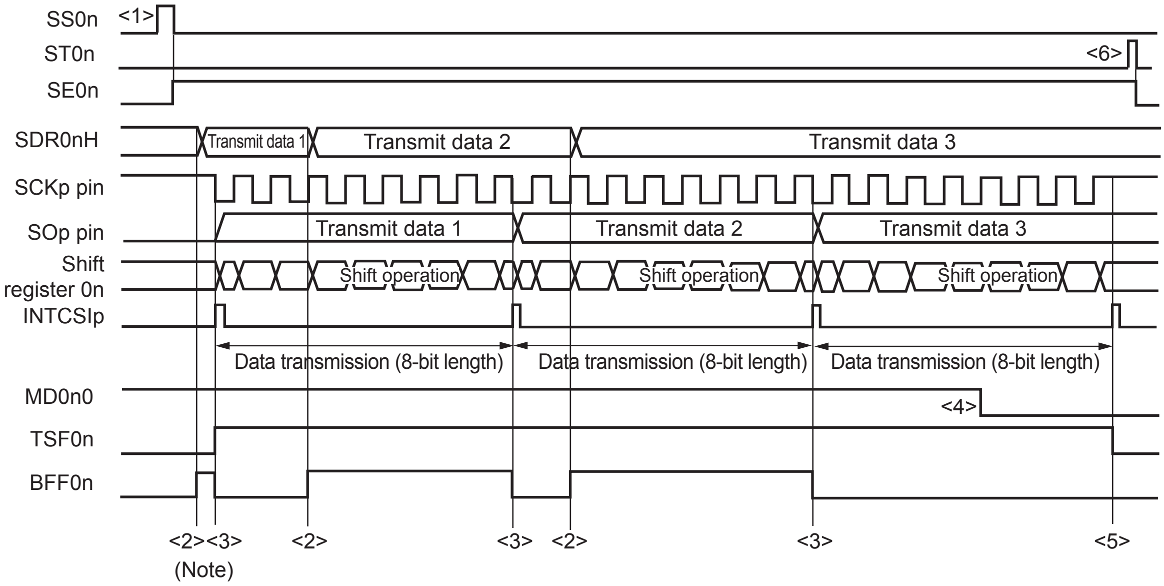
Remark n = 0, p: CSI number (p = 00)

Figure 10-29. Flowchart of Master Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 10-30. Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAP0n = 0, CKP0n = 0)

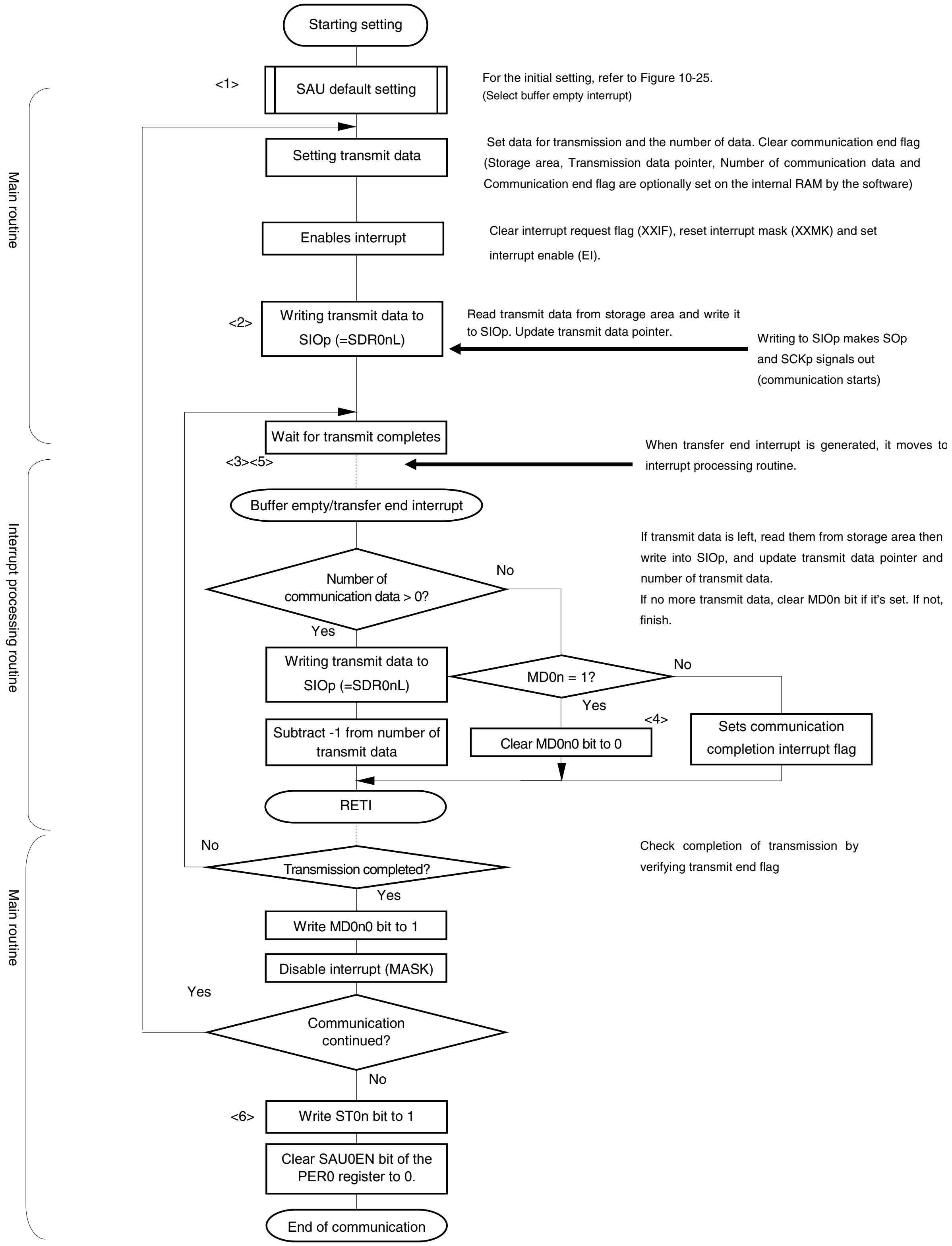


Note If transmit data is written to the SDR0nL register while the BFF0n bit of serial status register 0n (SSR0n) is 1 (valid data is stored in serial data register 0n (SDR0nL)), the transmit data is overwritten.

Caution The MD0n0 bit of serial mode register 0n (SMR0nL) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark n = 0, p: CSI number (p = 00)

Figure 10-31. Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 10-30 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

10.5.2 Master reception

Master reception is that the R7F0C80112ESP, R7F0C80212ESP output a transfer clock and receives data from other device.

3-Wire Serial I/O	CSI00
Target channel	Channel 0 of SAU0
Pins used	$\overline{\text{SCK00}}$, SI00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVF0n) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{CLK}/4$ [Hz] (SDR0nH[7:1] = 1 or more) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]Note 2
Data phase	Selectable by the DAP0n bit of the SCR0nH register • DAP0n = 0: Data input starts at the start of the operation of the serial clock. • DAP0n = 1: Data input starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKP0n bit of the SCR0nH register • CKP0n = 0: Non-inversion • CKP0n = 1: Inverted
Data direction	MSB or LSB first

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 21 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{CLK}: System clock frequency

2. n = 0

(1) Register setting

Figure 10-32. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol: SMR0nH

7	6	5	4	3	2	1	0
CKS0n	CCS0n						STS0n
0/1	0	0	0	0	0	0	0

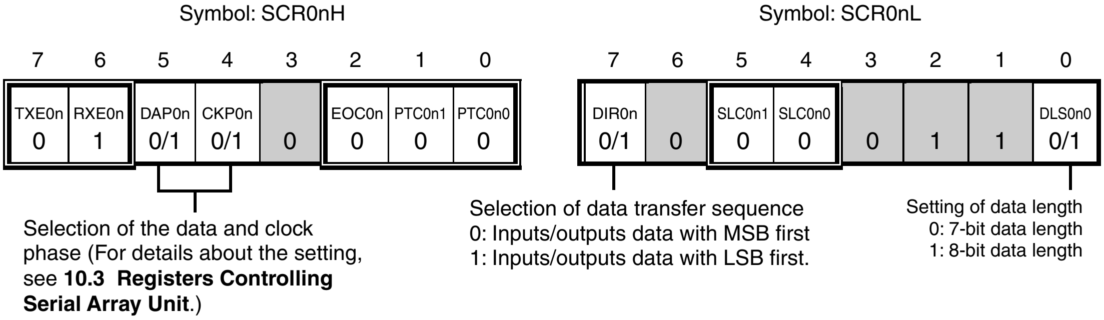
Symbol: SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)



Symbol: SCR0nH

7	6	5	4	3	2	1	0
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
0	1	0/1	0/1	0	0	0	0

Symbol: SCR0nL

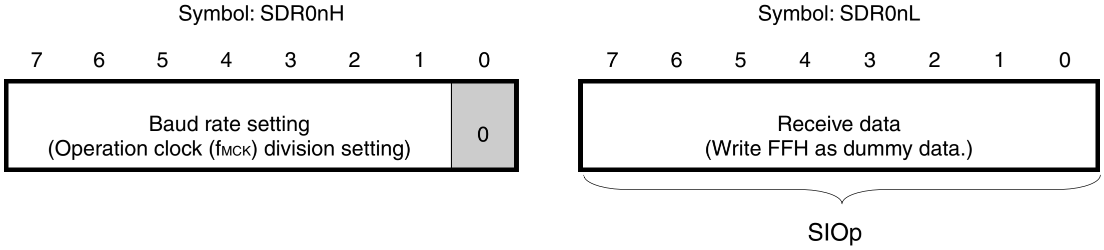
7	6	5	4	3	2	1	0
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **10.3 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 0n (SDR0nH, SDR0nL)



Symbol: SDR0nH

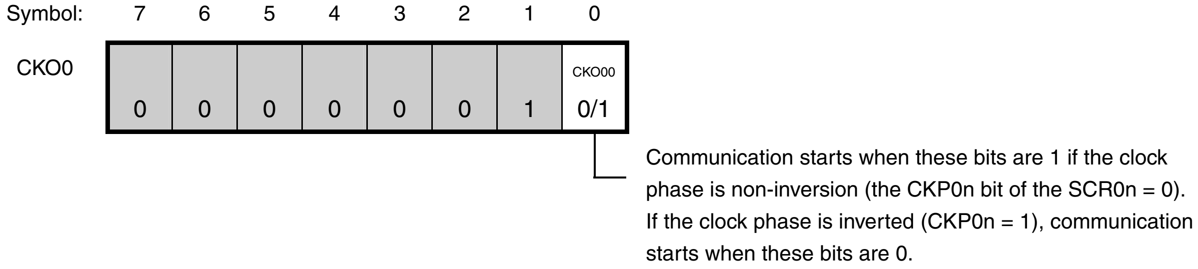
7	6	5	4	3	2	1	0
Baud rate setting (Operation clock (f_{MCK}) division setting)							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Receive data (Write FFH as dummy data.)							

SIOp

(d) Serial clock output register 0 (CKO0) … Sets only the bits of the target channel.

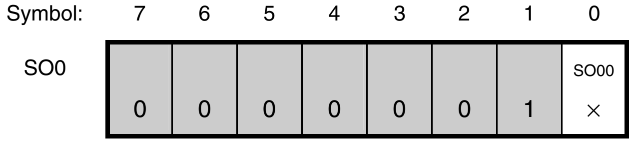


Symbol: CKO0

7	6	5	4	3	2	1	0
							CKO00
0	0	0	0	0	0	1	0/1

Communication starts when these bits are 1 if the clock phase is non-inversion (the CKP0n bit of the SCR0n = 0). If the clock phase is inverted (CKP0n = 1), communication starts when these bits are 0.

(e) Serial output register 0 (SO0) …The register that not used in this mode.



Symbol: SO0

7	6	5	4	3	2	1	0
							SO00
0	0	0	0	0	0	1	×

Figure 10-32. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00) (2/2)

(f) Serial output enable register 0 (SOE0) … The register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	SOE00 ×

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0	SS00 0/1

Remarks 1. n = 0, p: CSI number (p = 00)

2. ☐: Setting is fixed in the CSI master transmission mode, ▨: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 10-33. Initial Setting Procedure for Master Reception

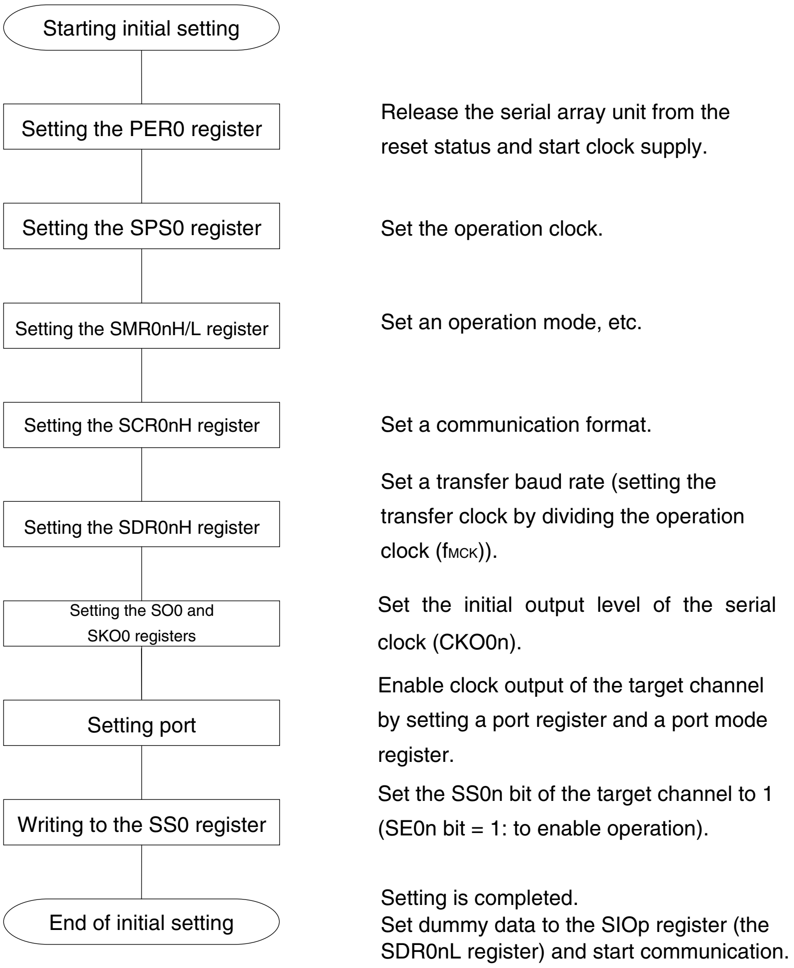


Figure 10-34. Procedure for Stopping Master Reception

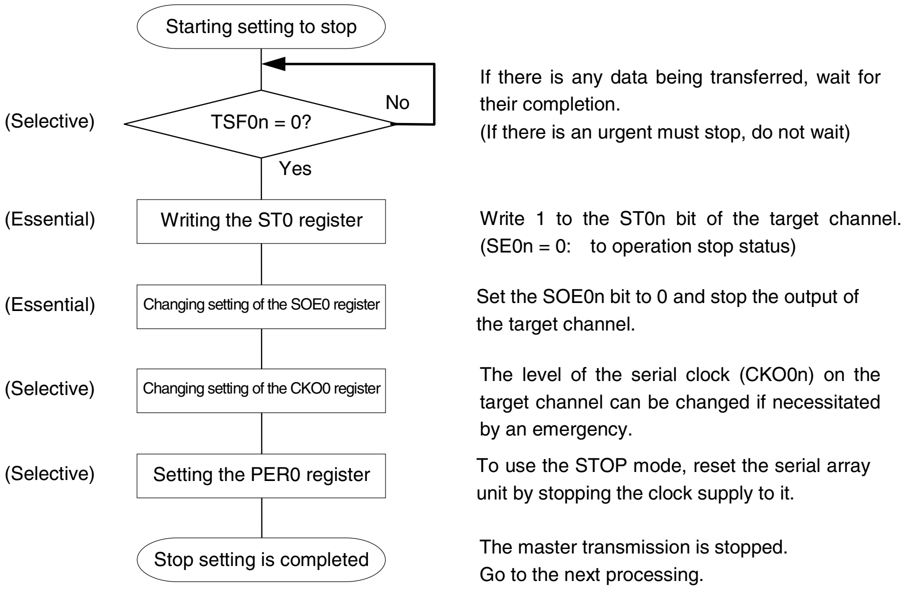
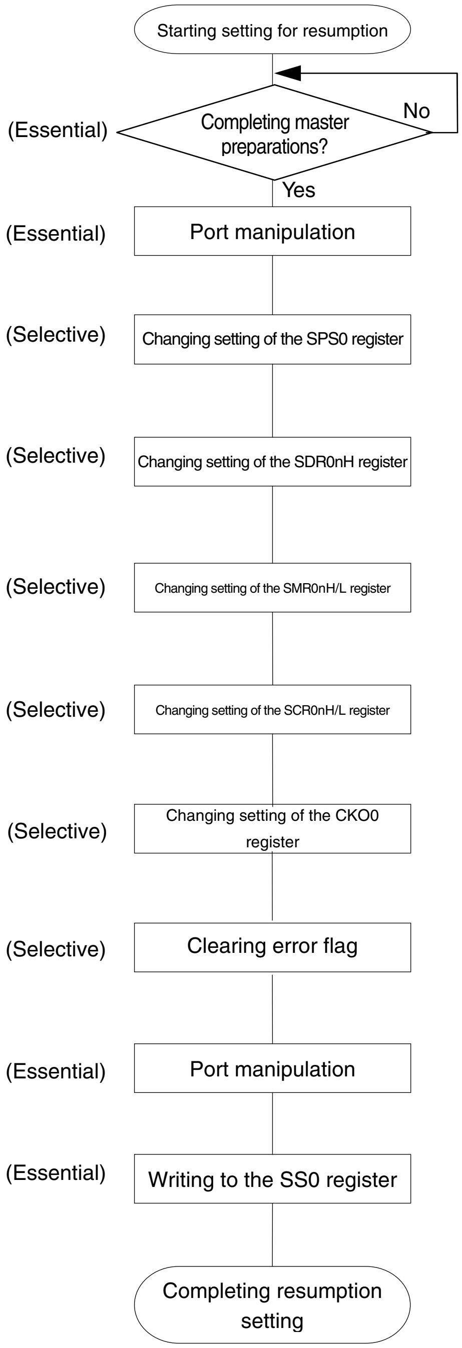


Figure 10-35. Procedure for Resuming Master Reception



Wait until stop the communication target (slave) or communication operation completed

Disable clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register 0n (SMR0nH/L) setting.

Re-set the register to change serial communication operation setting register 0n (SCR0nH/L) setting.

Set the initial output level of the serial clock (CKO0n).

If the OVF flag remain set, clear this using serial flag clear trigger register 0n (SIR0n).

Enable clock output of the target channel by setting a port register and a port mode register.

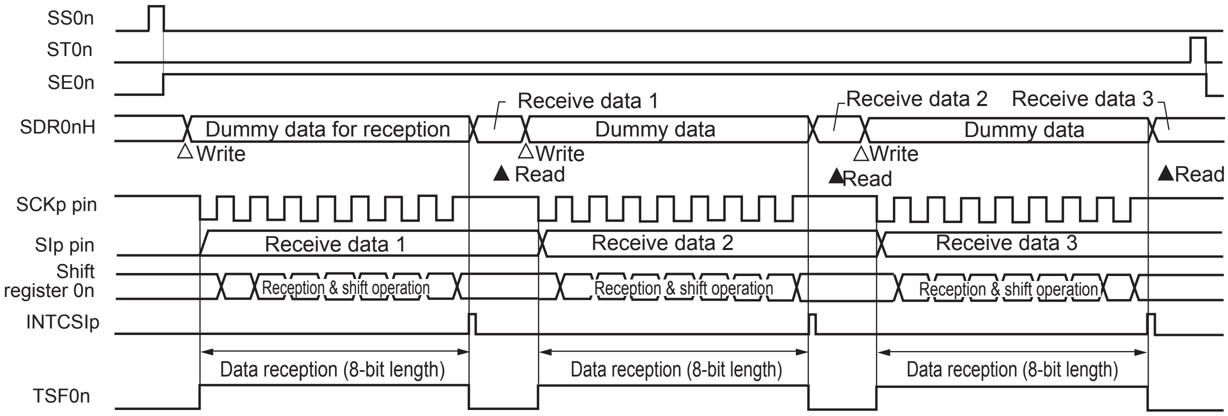
Set the SS0n bit of the target channel to 1 (SE0n bit = 1: to enable operation).

Setting is completed
Sets dummy data to the SIOp register (the SDR0nL register) and start communication.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

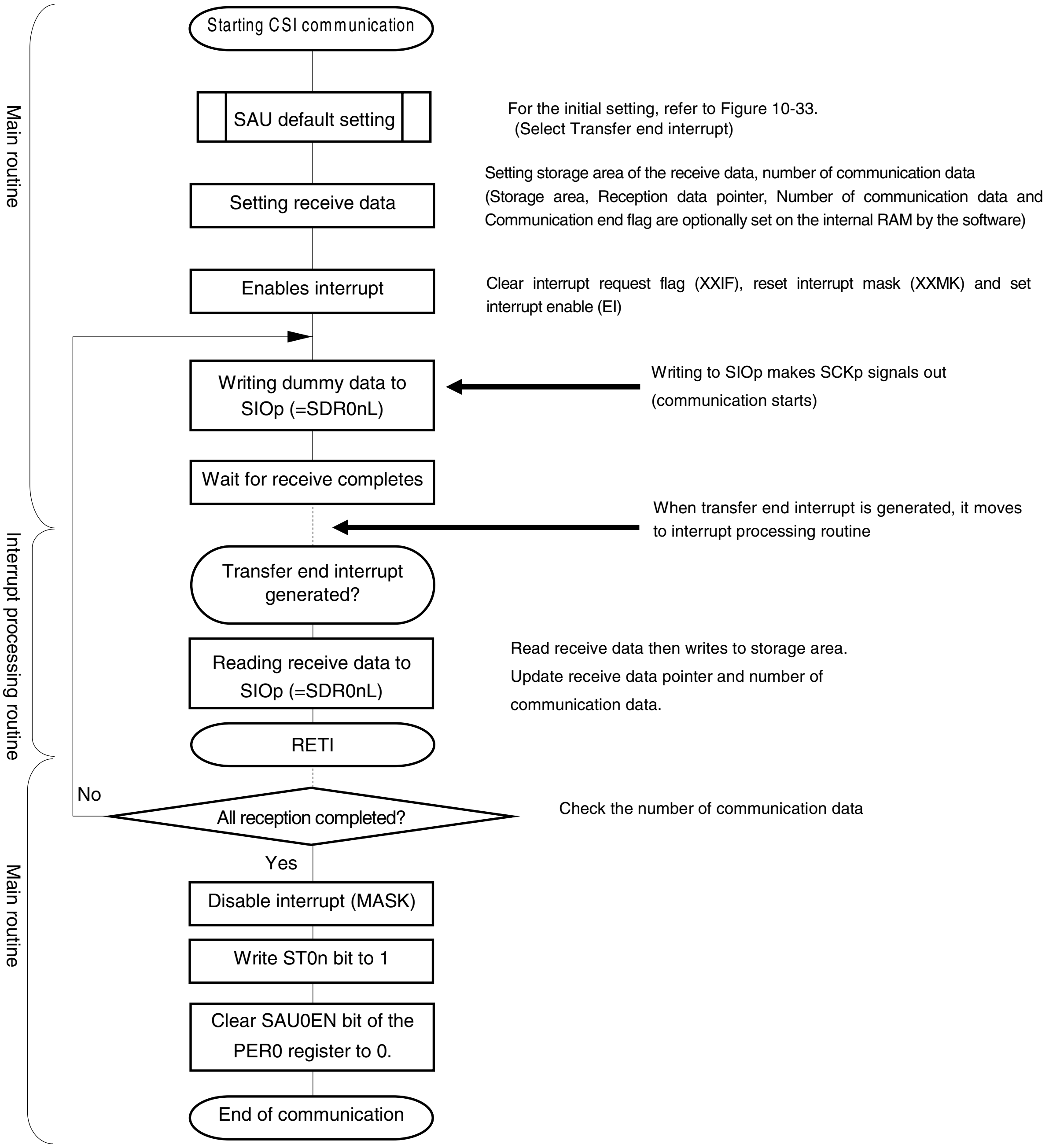
(3) Processing flow (in single-reception mode)

Figure 10-36. Timing Chart of Master Reception (in Single-Reception Mode) (Type 1: DAP0n = 0, CKP0n = 0)



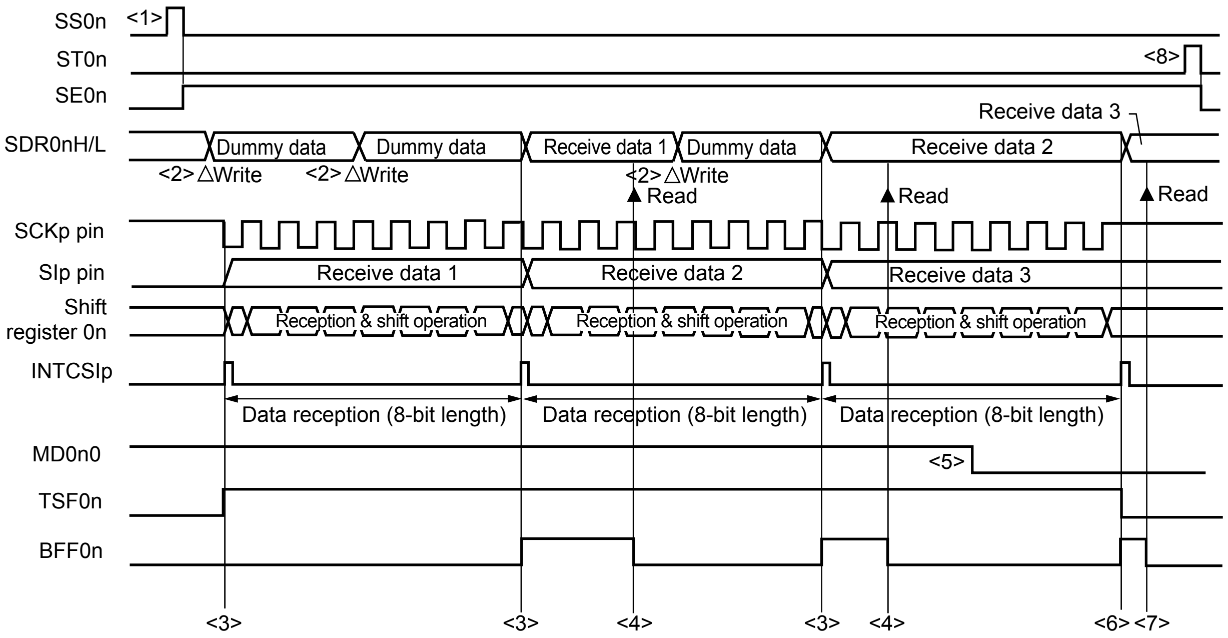
Remark n = 0, p: CSI number (p = 00)

Figure 10-37. Flowchart of Master Reception (in Single-Reception Mode)



(4) Processing flow (in continuous reception mode)

Figure 10-38. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAP0n = 0, CKP0n = 0)

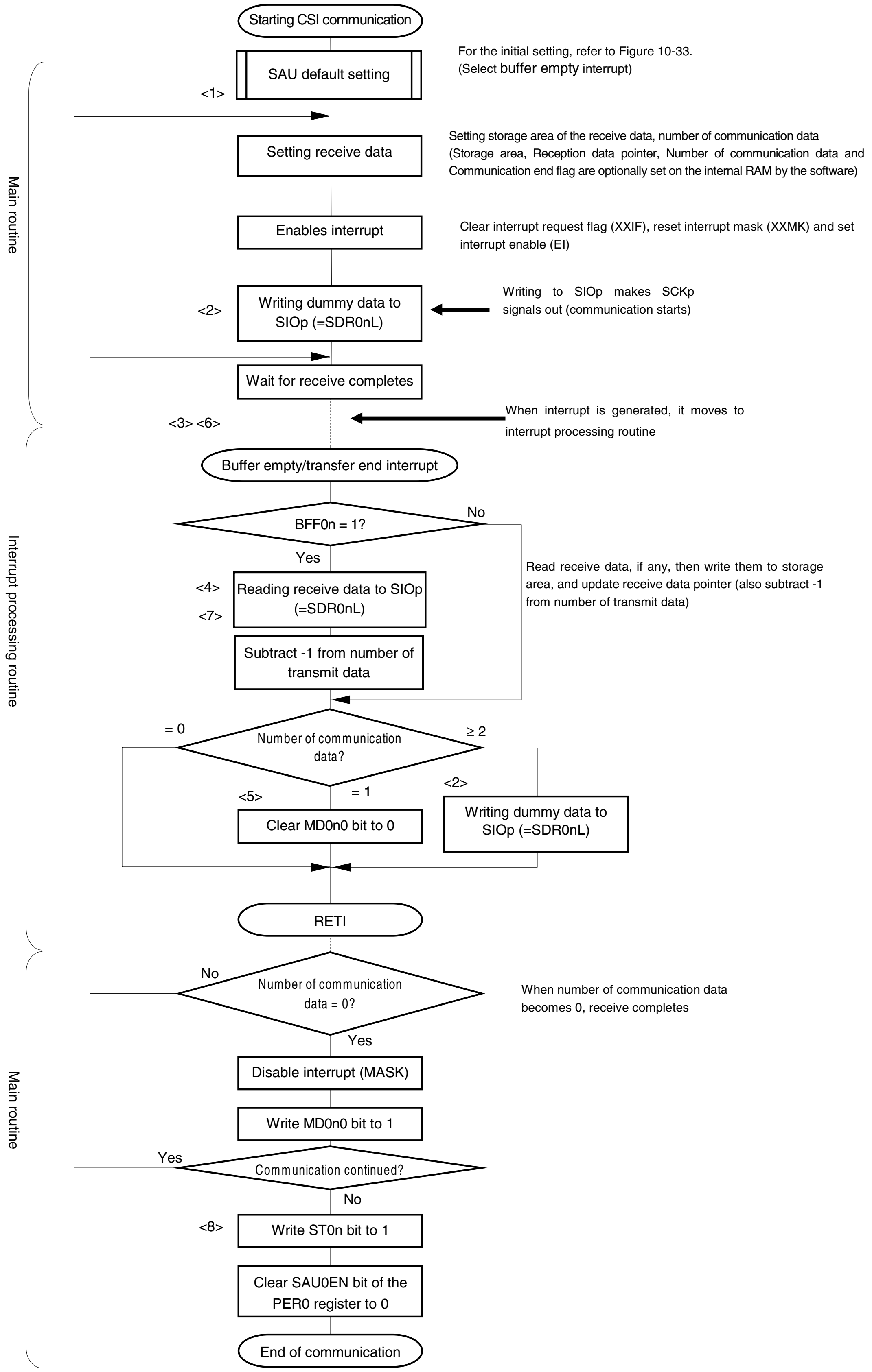


Caution The MD0n0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 10-39 Flowchart of Master Reception (in Continuous Reception Mode)**.

2. n = 0, p: CSI number (p = 00)

Figure 10-39. Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 10-38 Timing Chart of Master Reception (in Continuous Reception Mode)**.

10.5.3 Master transmission/reception

Master transmission/reception is that the R7F0C80112ESP, R7F0C80212ESP output a transfer clock and transmits/receives data to/from other device.

3-Wire Serial I/O	CSI00
Target channel	Channel 0 of SAU0
Pins used	$\overline{\text{SCK00}}$, SI00, SO00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVF0n) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{CLK}/4$ [Hz] (SDR0nH[7:1] = 1 or more) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]Note 2
Data phase	Selectable by the DAP0n bit of the SCR0nH register • DAP0n = 0: Data I/O starts at the start of the operation of the serial clock. • DAP0n = 1: Data I/O starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKP0n bit of the SCR0nH register • CKP0n = 0: Non-inversion • CKP0n = 1: Inverted
Data direction	MSB or LSB first

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 21 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{CLK}: System clock frequency

2. n = 0

(1) Register setting

Figure 10-40. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol:SMR0nH

7	6	5	4	3	2	1	0
CKS0n	CCS0n						STS0n
0/1	0	0	0	0	0	0	0

Symbol:SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol:SCR0nH

7	6	5	4	3	2	1	0
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
1	1	0/1	0/1	0	0	0	0

Symbol:SCR0nL

7	6	5	4	3	2	1	0
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **10.3 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 0n (SDR0nH, SDR0nL)

7	6	5	4	3	2	1	0
Baud rate setting (Operation clock (f_{MCK}) division setting)							0

7	6	5	4	3	2	1	0
Transmit data setting/receive data register							

SIOp

(d) Serial clock output register 0 (CKO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
CKO0								CKO00
	0	0	0	0	0	0	1	0/1

Communication starts when these bits are 1 if the clock phase is non-inversion (the CKP0n bit of the SCR0n = 0). If the clock phase is inverted (CKP0n = 1), communication starts when these bits are 0.

(e) Serial output register 0 (SO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
SO0								SO00
	0	0	0	0	0	0	1	0/1

Figure 10-40. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00) (2/2)

(f) Serial output enable register 0 (SOE0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	SOE00 0/1

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0	SS00 0/1

Remarks 1. n = 0, p: CSI number (p = 00)

2. ☐: Setting is fixed in the CSI master transmission/reception mode

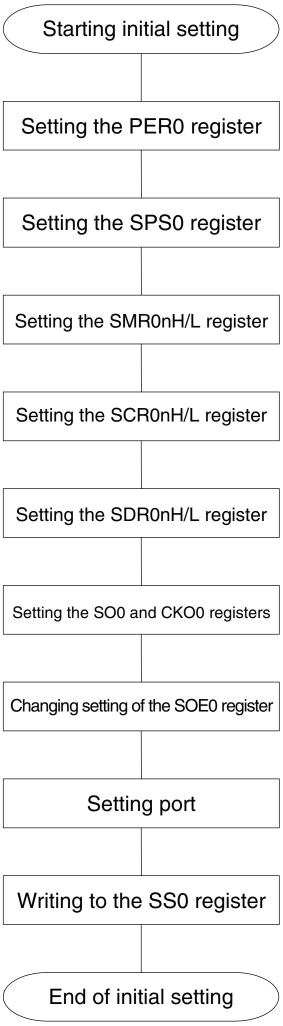
▨ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 10-41. 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 (CKO0n) and serial data (SO0n).

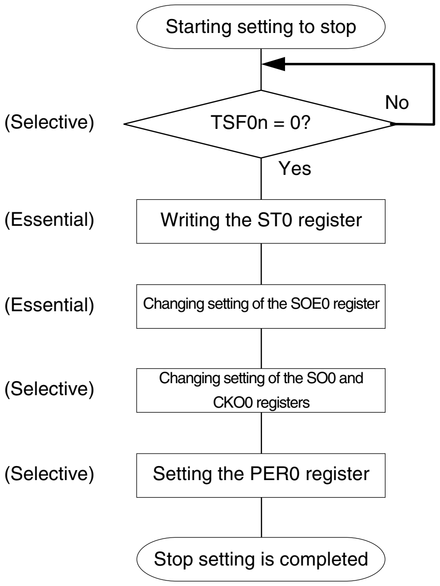
Set the SOE0n bit to 1 and enable data output of the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SS0n bit of the target channel to 1 (SE0n bit = 1: to enable operation).

Setting is completed
Set transmit data to the SIOp register (the SDR0nL register) and start communication.

Figure 10-42. 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 ST0n bit of the target channel.
(SE0n = 0: to operation stop status)

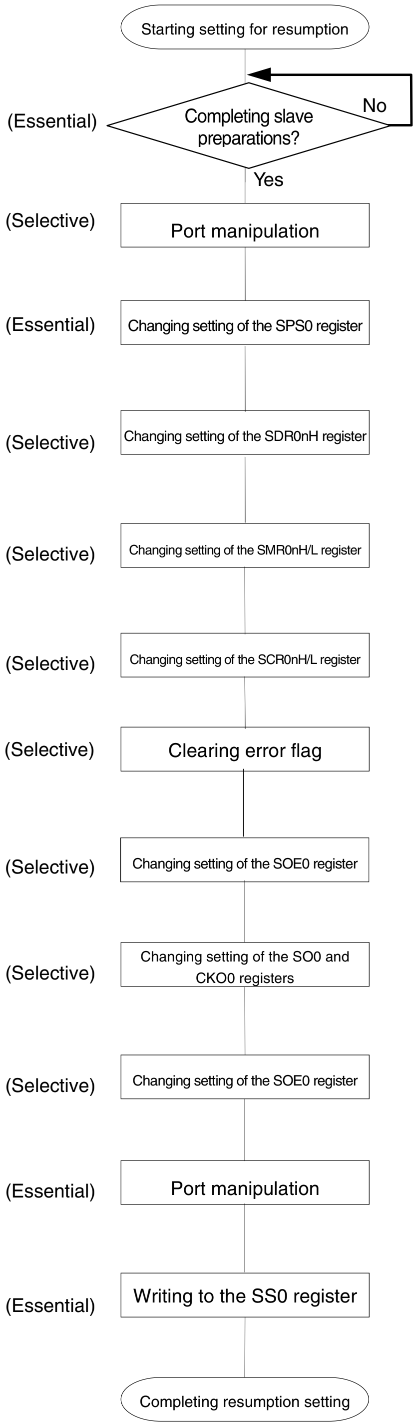
Set the SOE0n bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKO0n) and serial data (SO0n) 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 10-43. Procedure for Resuming Master Transmission/Reception



Wait until stop the communication target (slave) or communication operation completed

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register 0n (SMR0nH/L) setting.

Re-set the register to change serial communication operation setting register 0n (SCR0nH/L) setting.

If the FEF, PEF, and OVF flags remain set, clear them using serial flag clear trigger register 0n (SIR0n).

Set the SOE0n bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKO0n) and serial data (SO0n).

Set the SOE0n bit to 1 and enable output from the target channel.

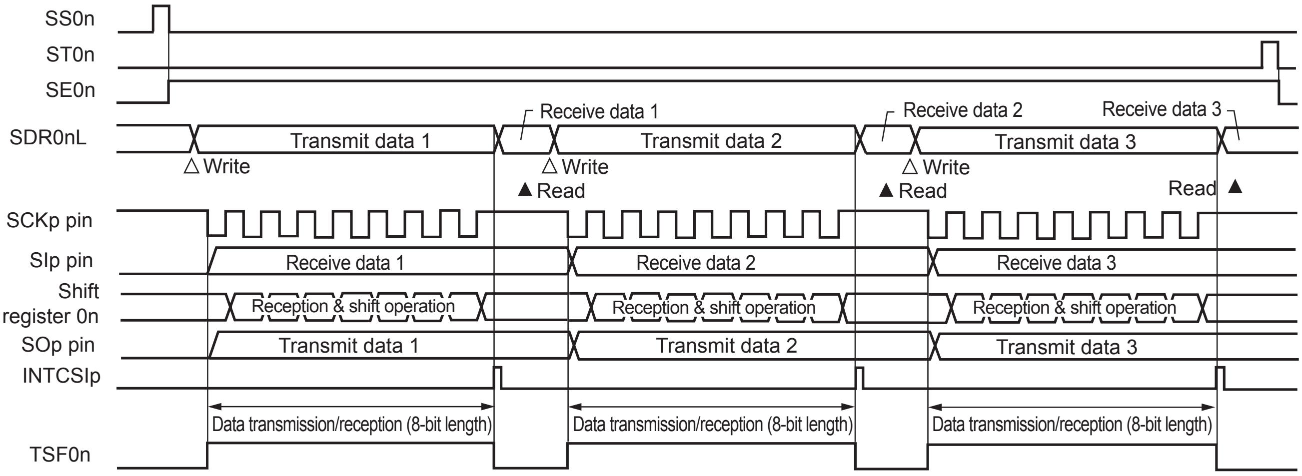
Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SS0n bit of the target channel to 1 (SE0n = 1: to enable operation).

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.

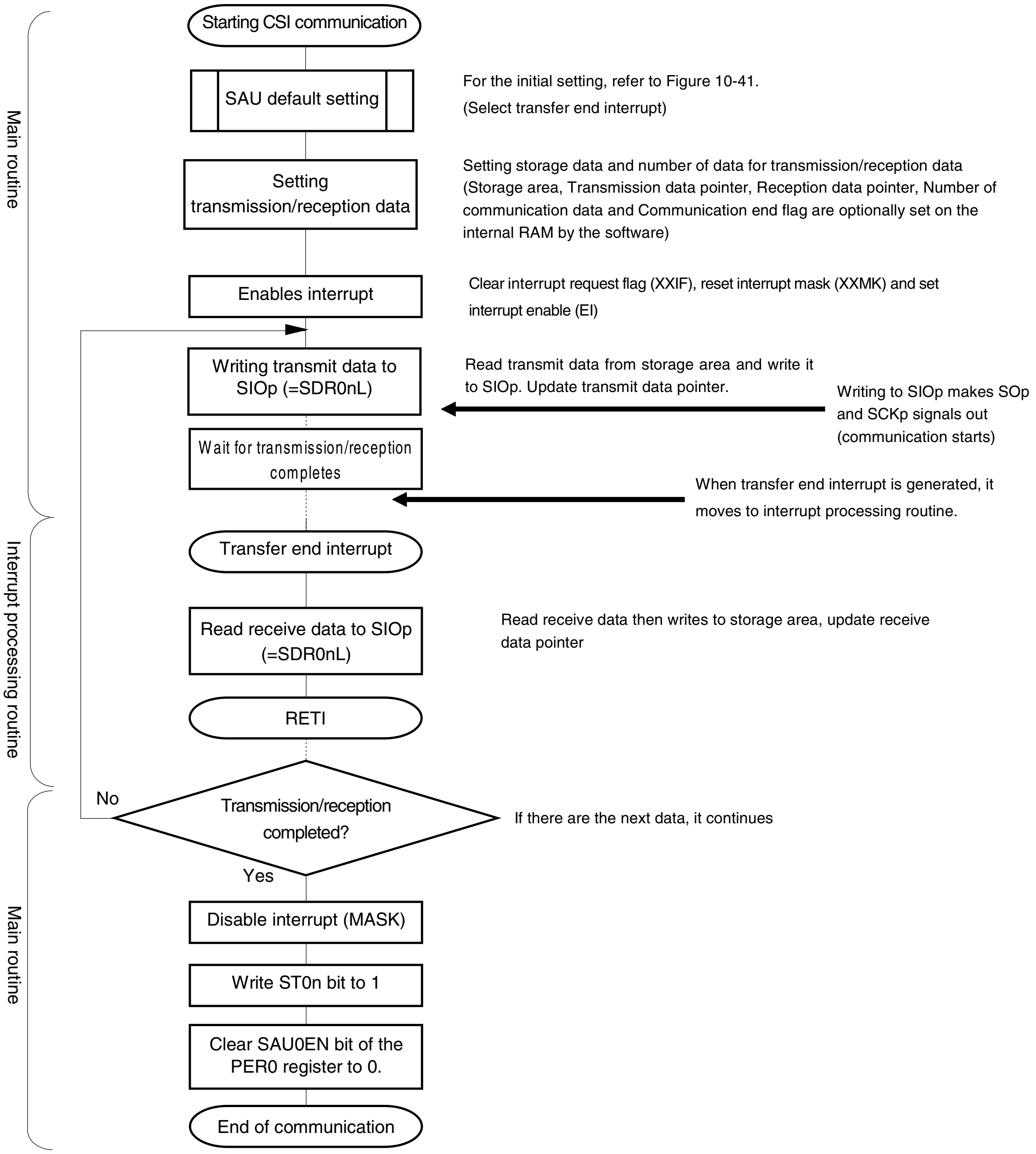
(3) Processing flow (in single-transmission/reception mode)

Figure 10-44. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAP0n =0, CKP0n = 0)



Remark n = 0, p: CSI number (p = 00)

Figure 10-45. Flowchart of Master Transmission/Reception (in Single-Transmission/Reception Mode)



For the initial setting, refer to Figure 10-41.
(Select transfer end interrupt)

Setting storage data and number of data for transmission/reception data (Storage area, Transmission data pointer, 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.

Writing to SIOp makes SOp and SCKp signals out (communication starts)

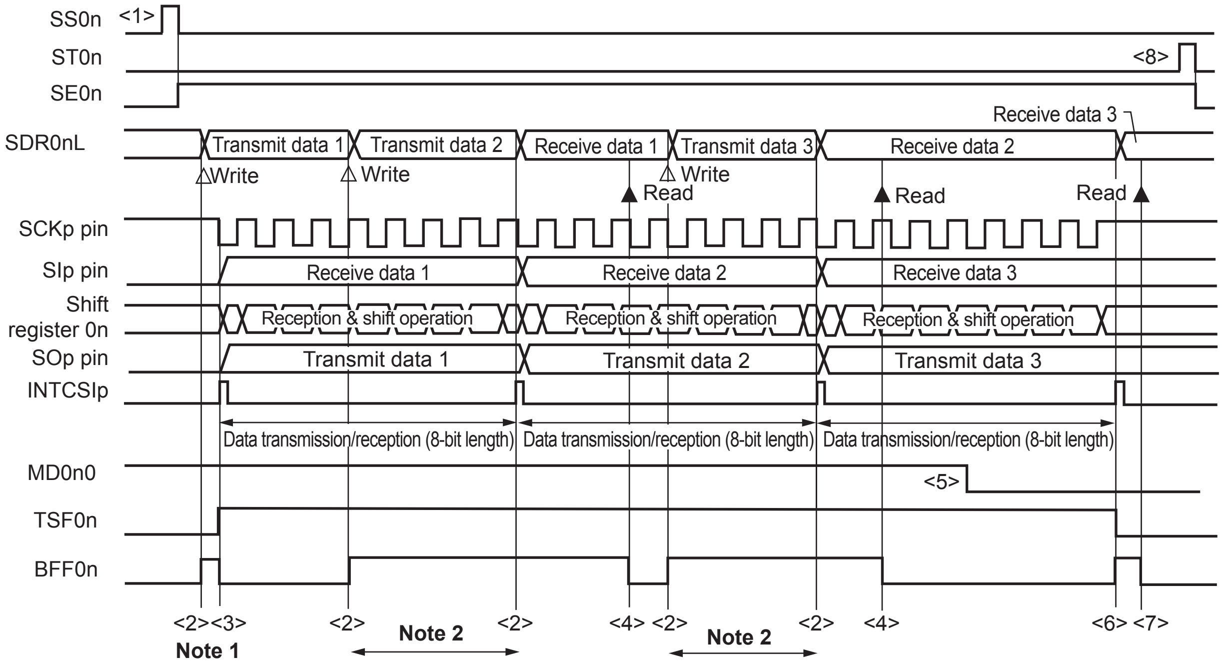
When transfer end interrupt is generated, it moves to interrupt processing routine.

Read receive data then writes to storage area, update receive data pointer

If there are the next data, it continues

(4) Processing flow (in continuous transmission/reception mode)

Figure 10-46. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAP0n = 0, CKP0n = 0)



Notes 1. If transmit data is written to the SDR0nL register while the BFF0n bit of serial status register 0n (SSR0n) is 1 (valid data is stored in serial data register 0n (SDR0nL)), the transmit data is overwritten.

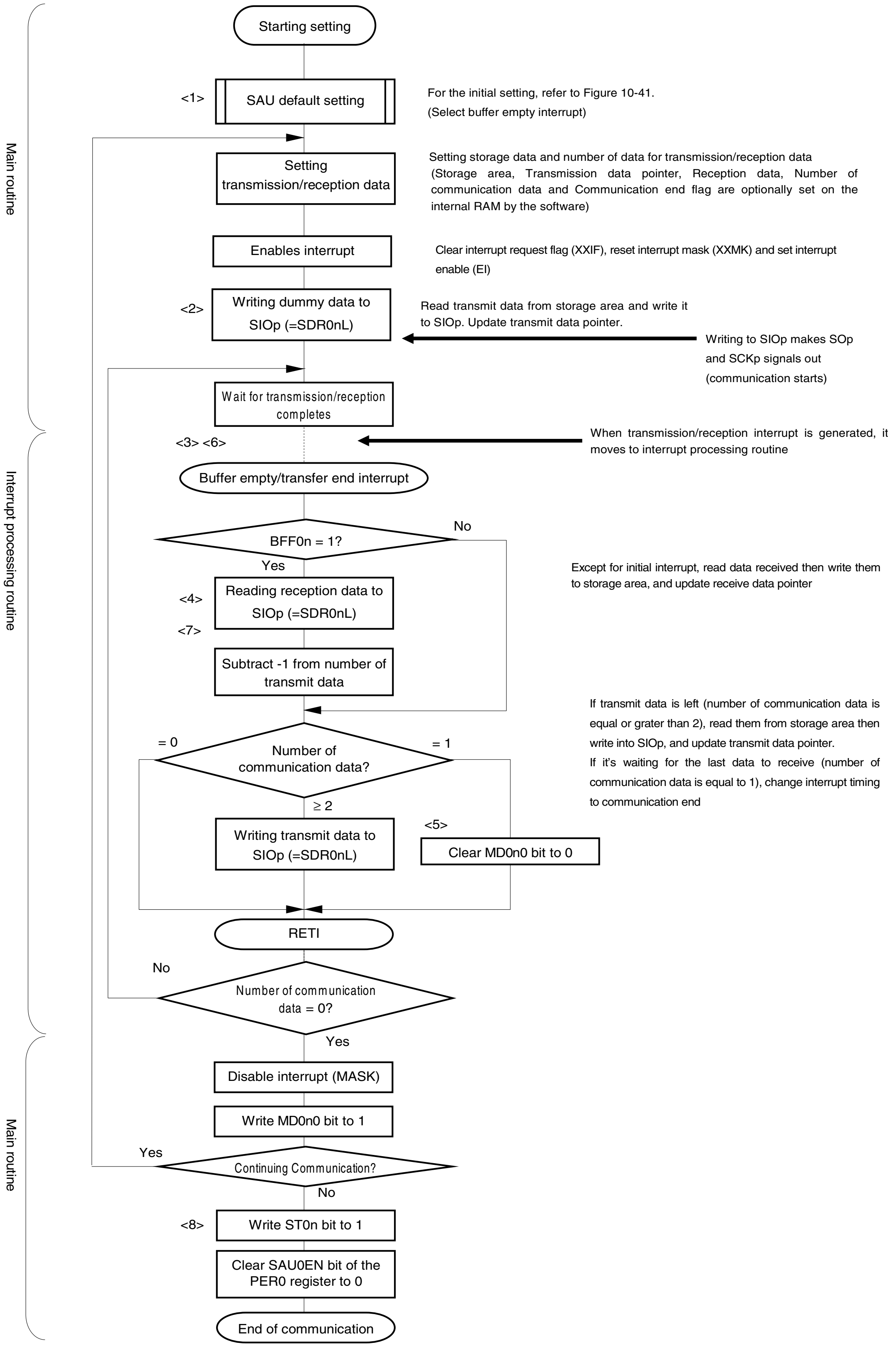
2. The transmit data can be read by reading the SDR0nL register during this period. At this time, the transfer operation is not affected.

Caution The MD0n0 bit of serial mode register 0n (SMR0nL) 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 10-47 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. n = 0, p: CSI number (p = 00)

Figure 10-47. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 10-46 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

10.5.4 Slave transmission

Slave transmission is that the R7F0C80112ESP, R7F0C80212ESP transmit data to another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00
Target channel	Channel 0 of SAU0
Pins used	$\overline{\text{SCK00}}$, SO00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVF0n) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{CLK}/6$ [Hz]^{Notes 1, 2}
Data phase	Selectable by the DAP0n bit of the SCR0nH register • DAP0n = 0: Data output starts at the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKP0n bit of the SCR0nH register • CKP0n = 0: Non-inversion • CKP0n = 1: Inverted
Data direction	MSB or LSB first

Notes 1. Because the external serial clock input to the $\overline{\text{SCK00}}$ pin is sampled internally and used, the fastest transfer rate is $f_{CLK}/6$ [Hz]. Set up the SPS0 register so that this external clock is at least $f_{SCK}/2$ as set by the SDR0nH register.

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 21 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{CLK}: System clock frequency

f_{SCK}: Serial clock frequency

2. n = 0

(1) Register setting

Figure 10-48. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol: SMR0nH

7	6	5	4	3	2	1	0
CKS0n	CCS0n						STS0n
0/1	1	0	0	0	0	0	0

Symbol: SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol: SCR0nH

7	6	5	4	3	2	1	0
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
1	0	0/1	0/1	0	0	0	0

Symbol: SCR0nL

7	6	5	4	3	2	1	0
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **10.3 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 0n (SDR0nH, SDR0nL)

Symbol: SDR0nH

7	6	5	4	3	2	1	0
0000000 Baud rate setting							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Transmit data setting							

SIOp

(d) Serial clock output register 0 (CKO0) … The Register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
CKO0								CKO00
	0	0	0	0	0	0	1	×

(e) Serial output register 0 (SO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
SO0								SO00
	0	0	0	0	0	0	1	0/1

Figure 10-48. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00) (2/2)

(f) Serial output enable register 0 (SOE0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	SOE00 0/1

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0	SS00 0/1

Remarks 1. n = 0 p: CSI number (p = 00)

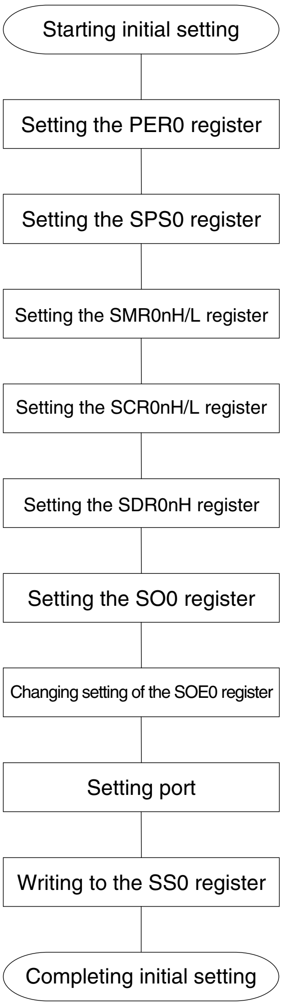
2. ☐: Setting is fixed in the CSI master transmission mode, ▨: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 10-49. Initial Setting Procedure for Slave Transmission



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 7 to 1 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SO0n).

Set the SOE0n 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 SS0n bit of the target channel to 1 (SE0n bit = 1 : to enable operation).

Initial setting is completed.
Set transmit data to the SIOp register (the SDR0nL register) and wait for a clock from the master.

Figure 10-50. Procedure for Stopping Slave Transmission

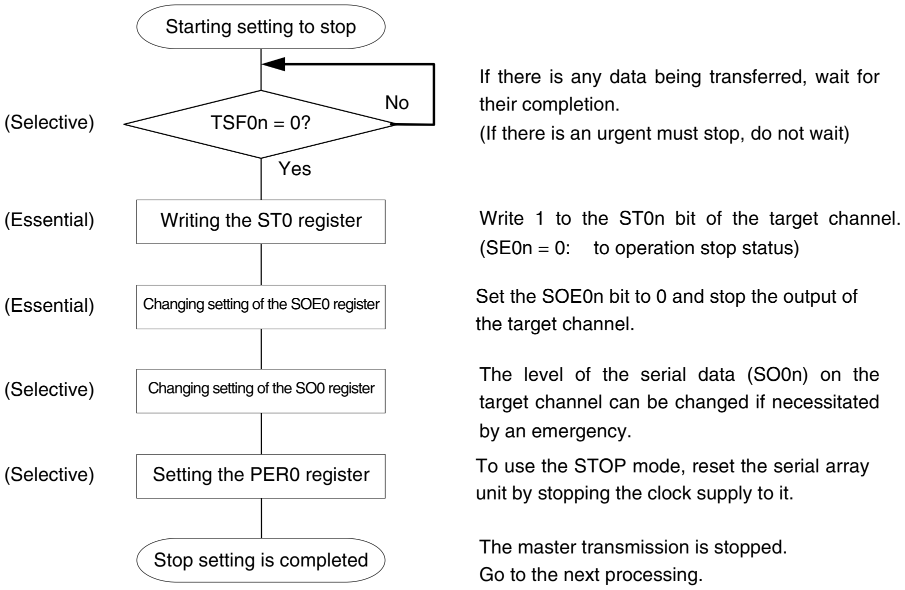
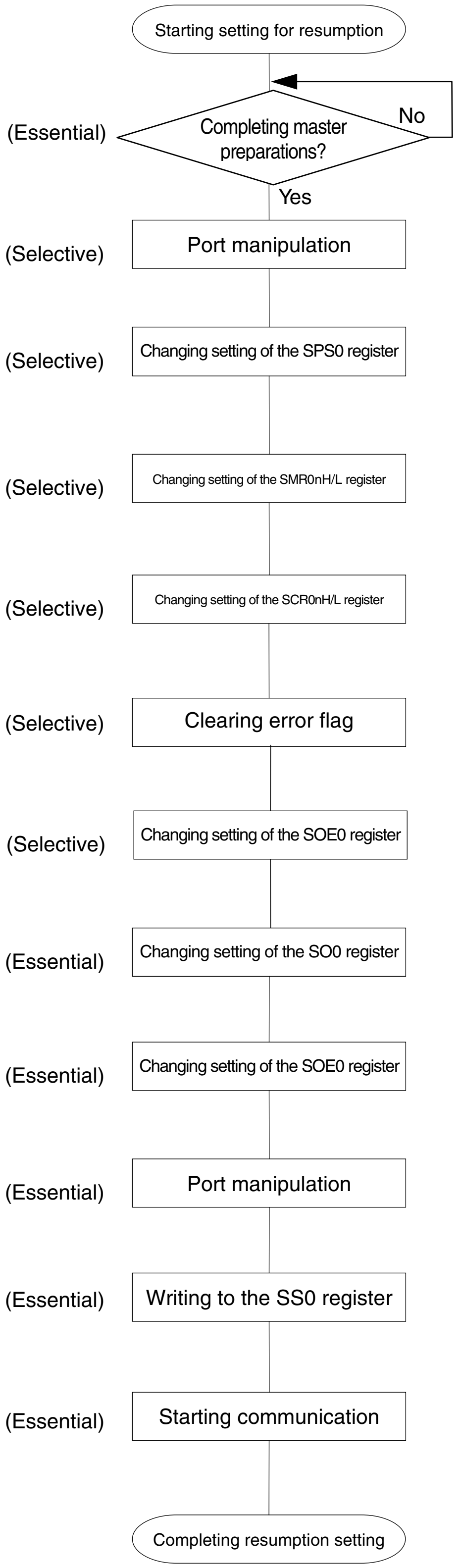


Figure 10-51. Procedure for Resuming Slave Transmission



Wait until stop the communication target (master)

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change serial mode register 0n (SMR0nH/L) setting.

Re-set the register to change serial communication operation setting register 0n (SCR0nH/L) setting.

If the OVF flag remain set, clear this using serial flag clear trigger register 0n (SIR0n).

Set the SOE0n bit to 0 to stop output from the target channel.

Set the initial output level of the serial data (SO0n).

Set the SOE0n 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.

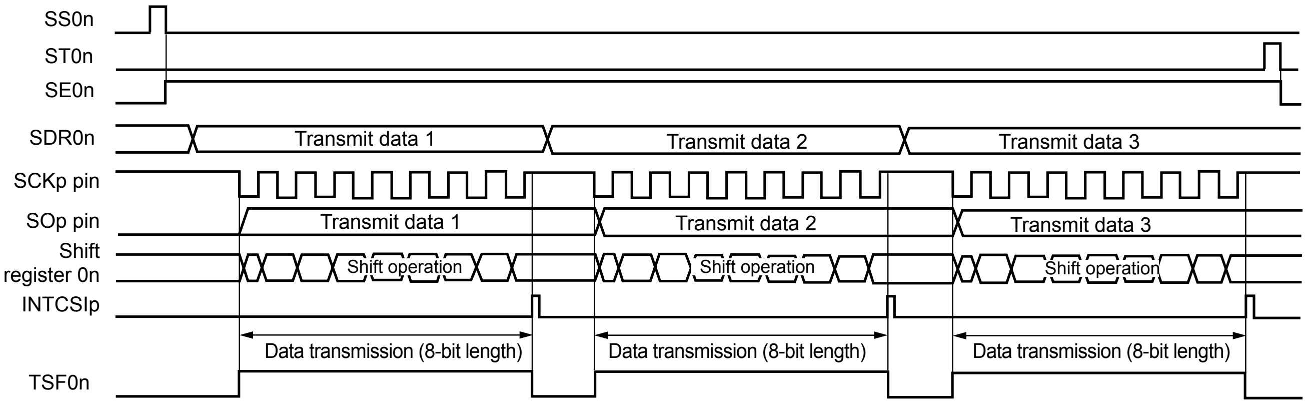
Set the SS0n bit of the target channel to 1 (SE0n = 1 : to enable operation).

Sets transmit data to the SIOp register (the SDR0nL 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.

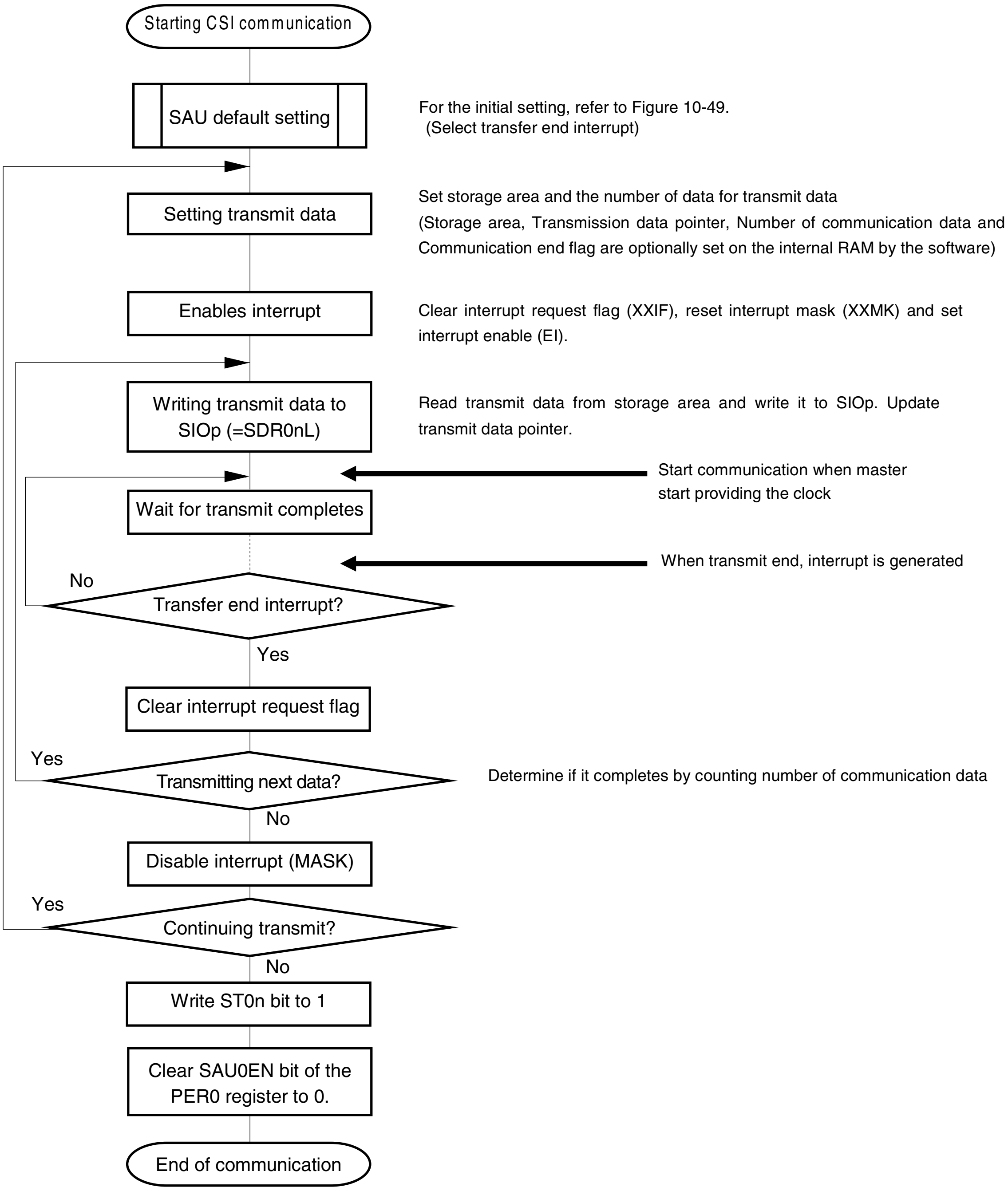
(3) Processing flow (in single-transmission mode)

Figure 10-52. Timing Chart of Slave Transmission (in Single-Transmission Mode) (Type 1: DAP0n = 0, CKP0n = 0)



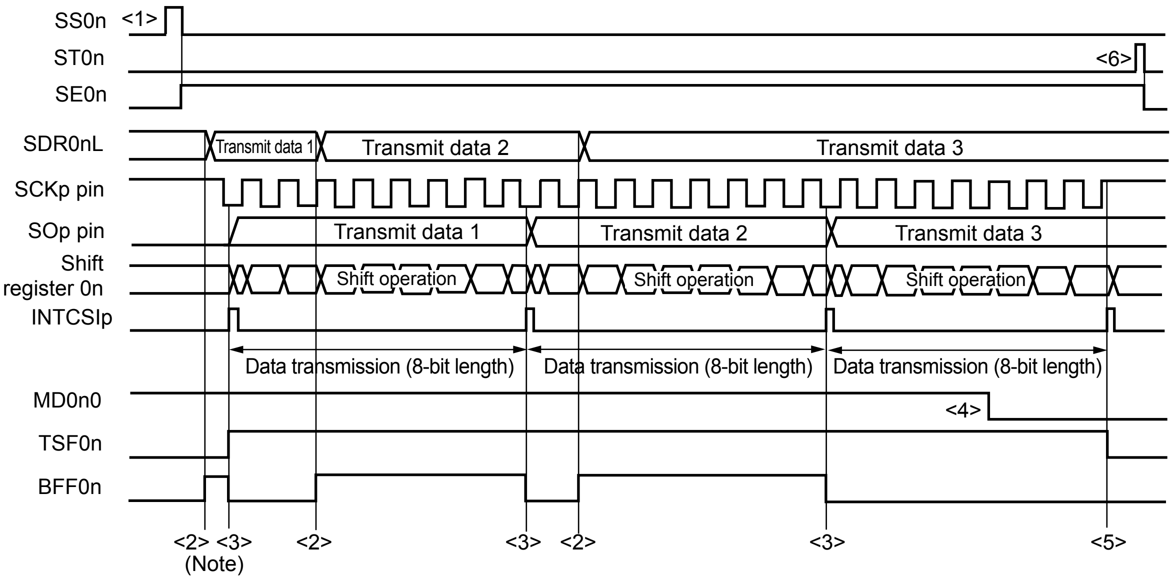
Remark n = 0, p: CSI number (p = 00)

Figure 10-53. Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 10-54. Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAP0n = 0, CKP0n = 0)

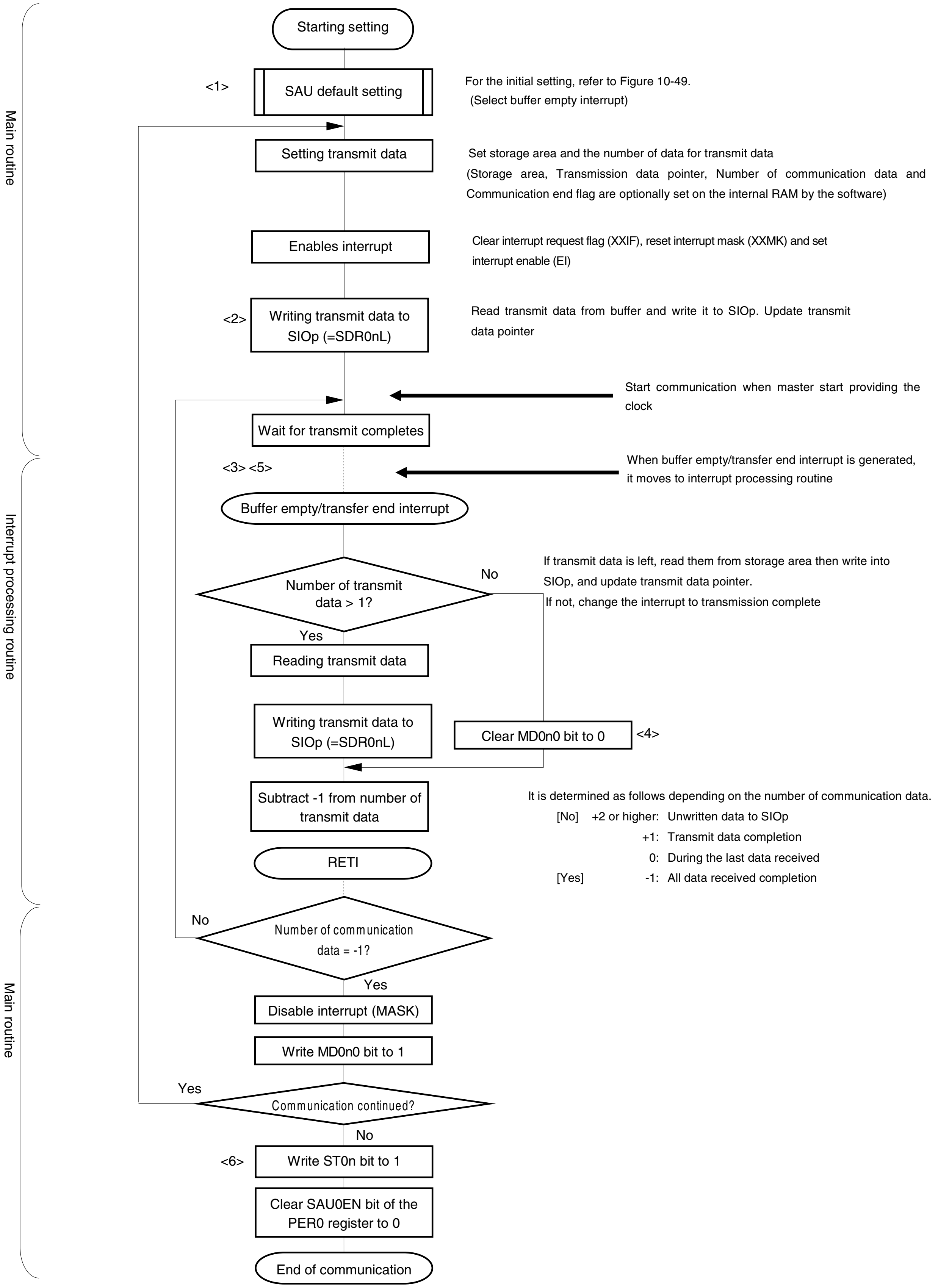


Note If transmit data is written to the SDR0nL register while the BFF0n bit of serial status register 0n (SSR0n) is 1 (valid data is stored in serial data register 0n (SDR0nL)), the transmit data is overwritten.

Caution The MD0n0 bit of serial mode register 0n (SMR0nL) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark n = 0, p: CSI number (p = 00)

Figure 10-55. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 10-54 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

10.5.5 Slave reception

Slave reception is that the R7F0C80112ESP, R7F0C80212ESP receive data from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00
Target channel	Channel 0 of SAU0
Pins used	$\overline{\text{SCK00}}$, SI00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVF0n) only
Transfer data length	7 or 8 bits
Transfer rate	Max. f_{CLK}/6 [Hz][Notes 1, 2]
Data phase	Selectable by the DAP0n bit of the SCR0nH register • DAP0n = 0: Data output starts at the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKP0n bit of the SCR0nH register • CKP0n = 0: Non-inversion • CKP0n = 1: Inverted
Data direction	MSB or LSB first

Notes 1. Because the external serial clock input to the $\overline{\text{SCK00}}$ pin is sampled internally and used, the fastest transfer rate is f_{CLK}/6 [Hz]. Set up the SPS0 register so that this external clock is at least f_{SCK}/2 as set by the SDR0nH register.

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 21 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{CLK}: System clock frequency

f_{SCK}: Serial clock frequency

2. n = 0

(1) Register setting

Figure 10-56. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol:SMR0nH

7	6	5	4	3	2	1	0
CKS0n	CCS0n						STS0n
0/1	1	0	0	0	0	0	0

Symbol:SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol: SCR0nH

7	6	5	4	3	2	1	0
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
0	1	0/1	0/1	0	0	0	0

Symbol: SCR0nL

7	6	5	4	3	2	1	0
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **10.3 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 0n (SDR0nH, SDR0nL)

Symbol: SDR0nH

7	6	5	4	3	2	1	0
0000000 Baud rate setting							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Receive data							

SIOp

(d) Serial clock output register 0 (CKO0) … The Register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
CKO0								CKO00
	0	0	0	0	0	0	1	×

(e) Serial output register 0 (SO0) … The Register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
SO0								SO00
	0	0	0	0	0	0	1	×

Figure 10-56. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00) (2/2)

(f) Serial output enable register 0 (SOE0) … The Register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	SOE00 ×

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0	SS00 0/1

Remarks 1. n = 0, p: CSI number (p = 00)

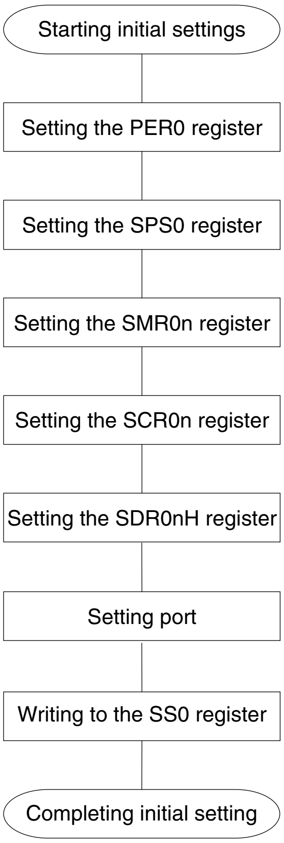
2. ☐ : Setting is fixed in the CSI master transmission mode, ▨ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 10-57. 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.

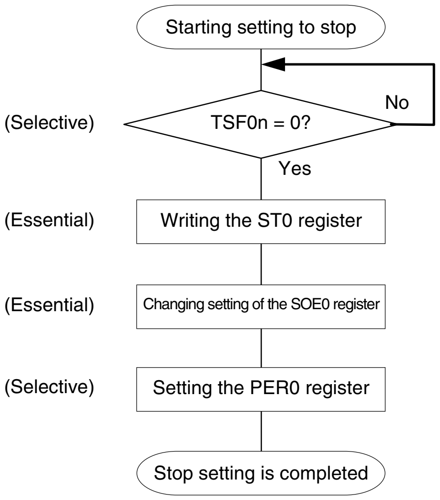
Set a communication format.

Set baud rate setting (bits 7 to 1) to 0000000B.

Enable data input and clock input of the target channel by setting a port register and a port mode register.

Set the SS0n bit of the target channel to 1 (SE0n bit = 1: to enable operation). Wait for a clock from the master.

Figure 10-58. 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)

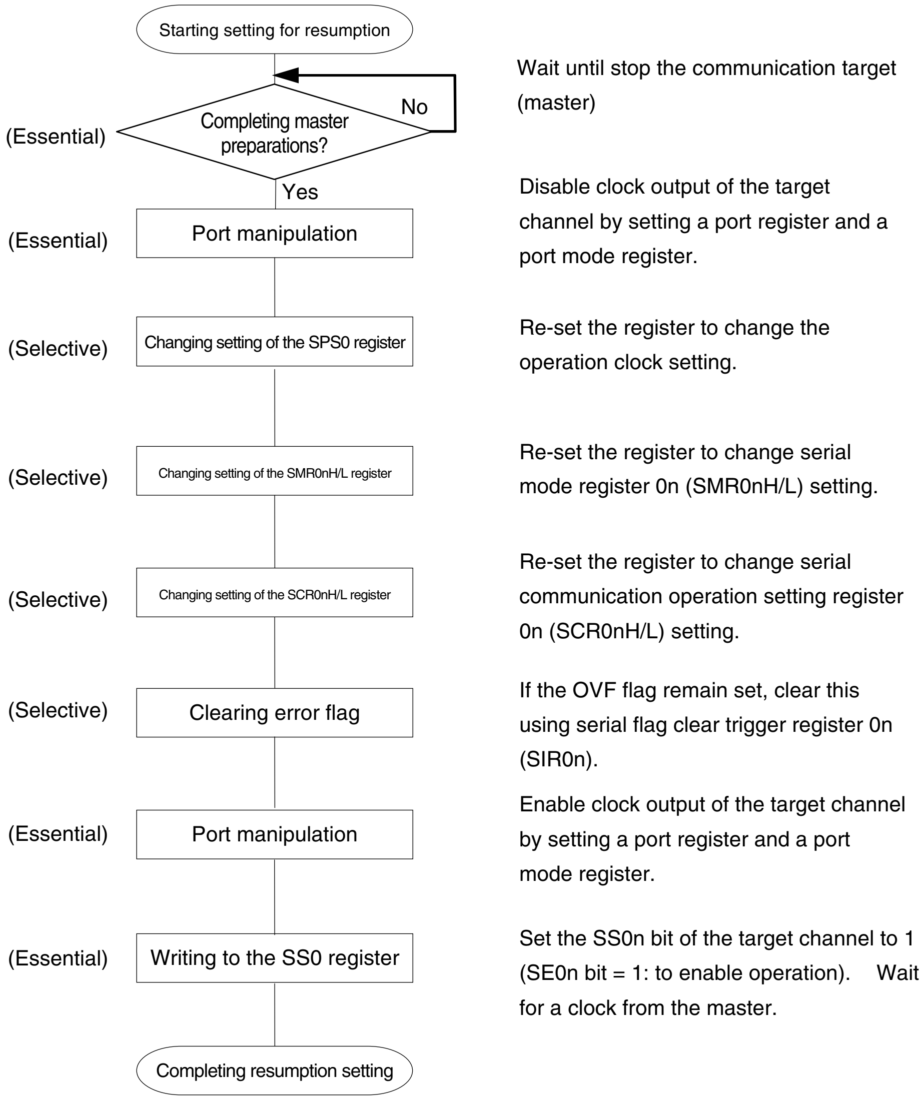
Write 1 to the ST0n bit of the target channel.
(SE0n = 0 : to operation stop status)

Set the SOE0n bit to 0 and stop the output of the target channel.

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.

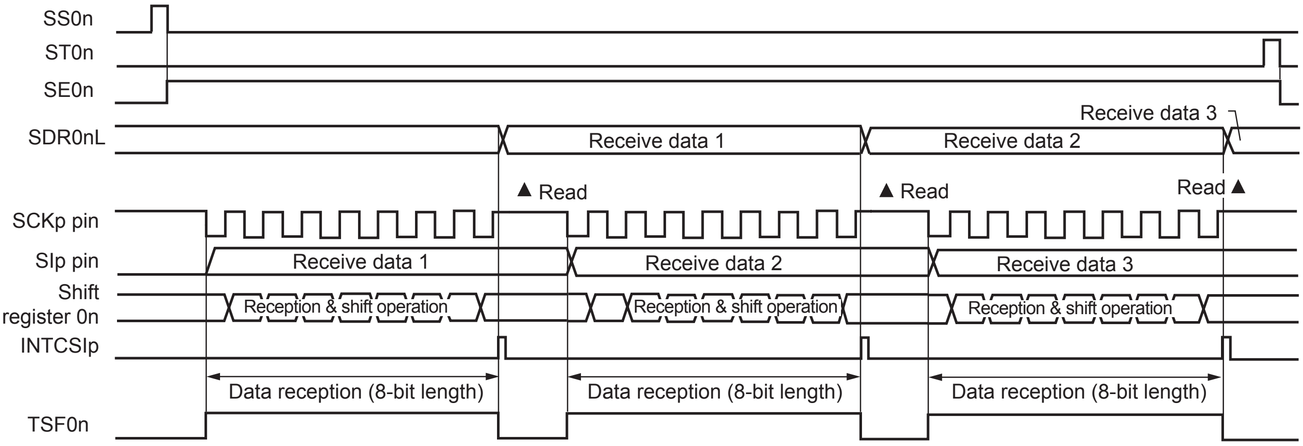
Figure 10-59. Procedure for Resuming Slave Reception



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

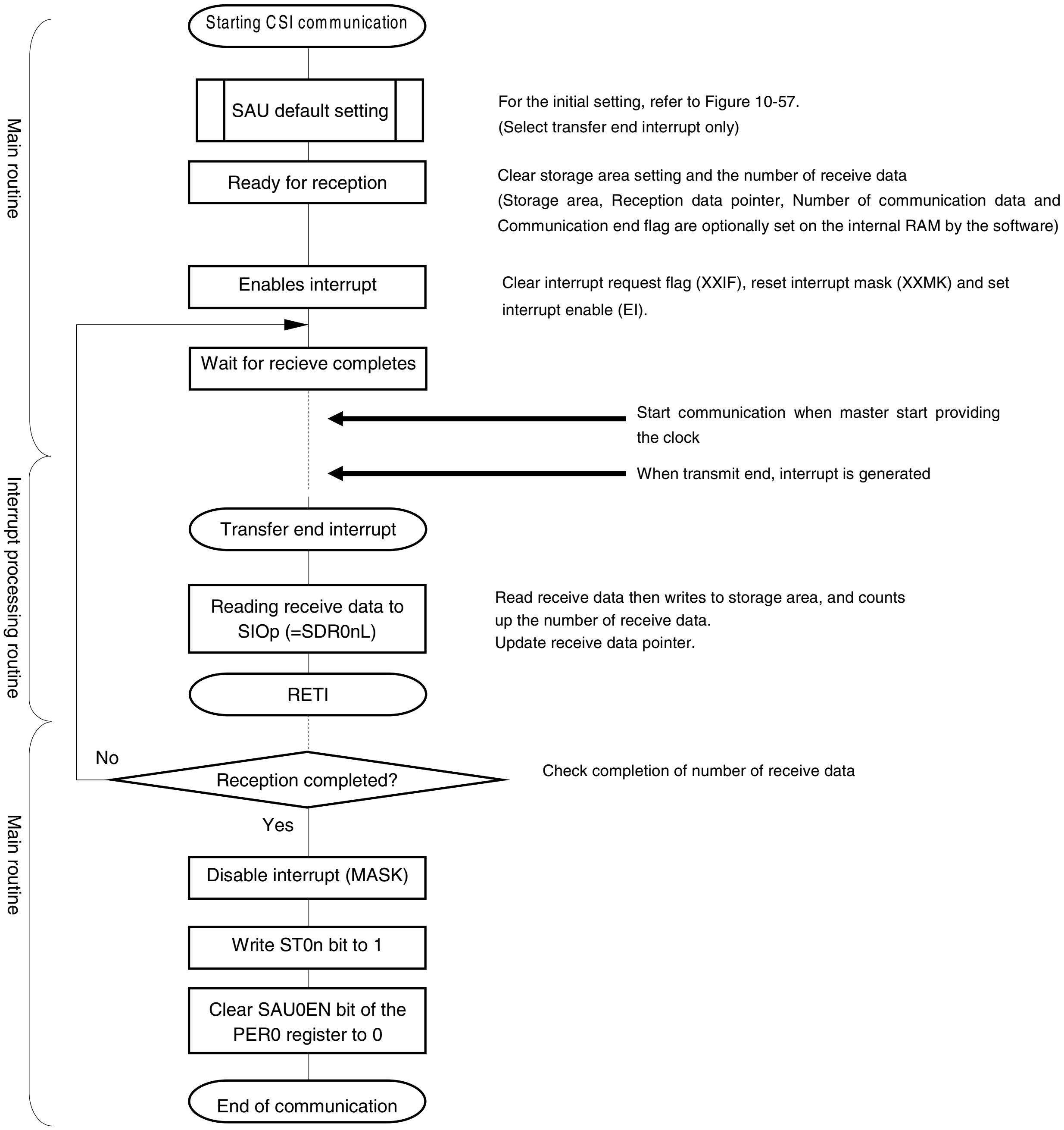
(3) Processing flow (in single-reception mode)

Figure 10-60. Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAP0n = 0, CKP0n = 0)



Remark n = 0, p: CSI number (p = 00)

Figure 10-61. Flowchart of Slave Reception (in Single-Reception Mode)



10.5.6 Slave transmission/reception

Slave transmission/reception is that the R7F0C80112ESP, R7F0C80212ESP transmit/receive data to/from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00
Target channel	Channel 0 of SAU0
Pins used	$\overline{\text{SCK00}}$, SI00, SO00
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVF0n) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{\text{CLK}}/6$ [Hz]$^{\text{Notes 1, 2,}}$
Data phase	Selectable by the DAP0n bit of the SCR0nH register • DAP0n = 0: Data output starts at the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKP0n bit of the SCR0nH register • CKP0n = 0: Non-inversion • CKP0n = 1: Inverted
Data direction	MSB or LSB first

Notes 1. Because the external serial clock input to the $\overline{\text{SCK00}}$ pin is sampled internally and used, the fastest transfer rate is $f_{\text{CLK}}/6$ [Hz]. Set up the SPS0 register so that this external clock is at least $f_{\text{SCK}}/2$ as set by the SDR0nH register.

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 21 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{CLK}: System clock frequency

f_{SCK}: Serial clock frequency

2. n = 0

(1) Register setting

Figure 10-62. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol: SMR0nH

7	6	5	4	3	2	1	0
CKS0n	CCS0n						STS0n
0/1	1	0	0	0	0	0	0

Operation clock (fMCK) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Symbol: SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0/1

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol: SCR0nH

7	6	5	4	3	2	1	0
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
1	1	0/1	0/1	0	0	0	0

Selection of the data and clock phase (For details about the setting, see **10.3 Registers Controlling Serial Array Unit.**)

Symbol: SCR0nL

7	6	5	4	3	2	1	
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0	0	0	1	1	0/1

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 0n (SDR0nH, SDR0nL)

Symbol: SDR0nH

7	6	5	4	3	2	1	0
0000000 Baud rate setting							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Transmit data setting/receive data register							

SIOp

(d) Serial clock output register 0 (CKO0) … The Register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
CKO0								CKO00
	0	0	0	0	0	0	1	×

(e) Serial output register 0 (SO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
SO0								SO00
	0	0	0	0	0	0	1	0/1

Figure 10-62. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00) (2/2)

(f) Serial output enable register 0 (SOE0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	SOE00 0/1

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0	SS00 0/1

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remarks 1. n = 0, p: CSI number (p = 00)

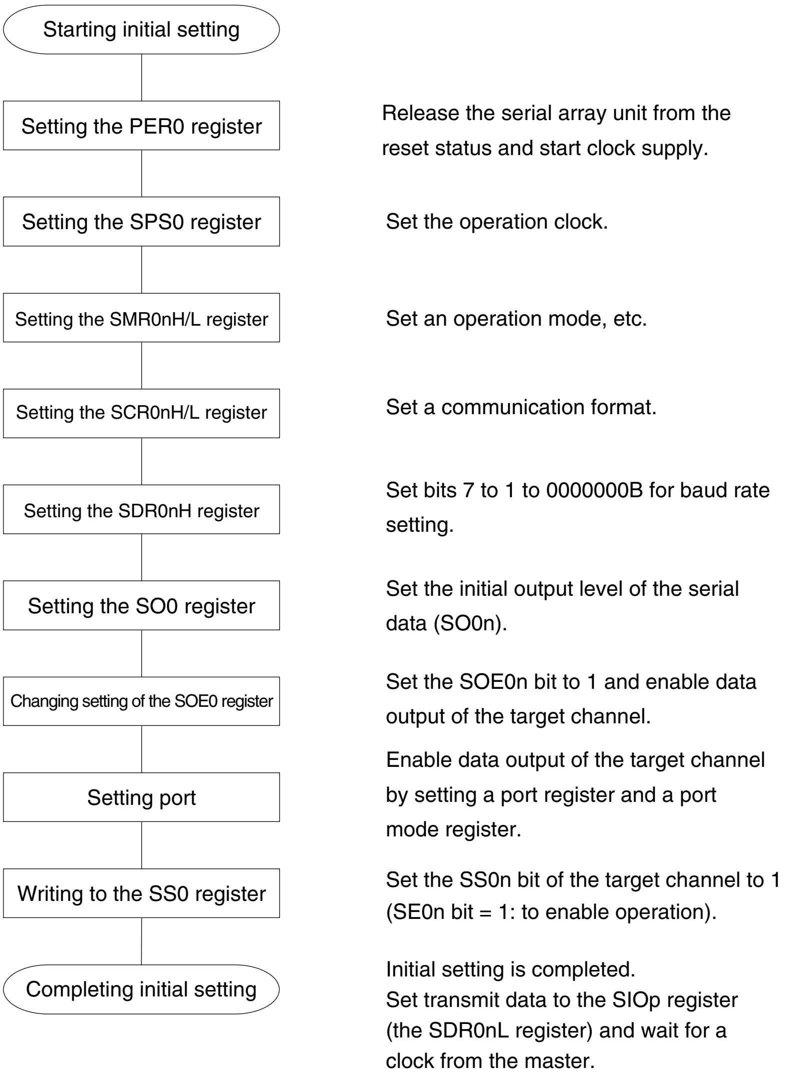
2. ☐: Setting is fixed in the CSI master transmission mode, ▨: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 10-63. 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 7 to 1 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SO0n).

Set the SOE0n 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 SS0n bit of the target channel to 1 (SE0n bit = 1: to enable operation).

Initial setting is completed.
Set transmit data to the SIOp register (the SDR0nL 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 10-64. Procedure for Stopping Slave Transmission/Reception

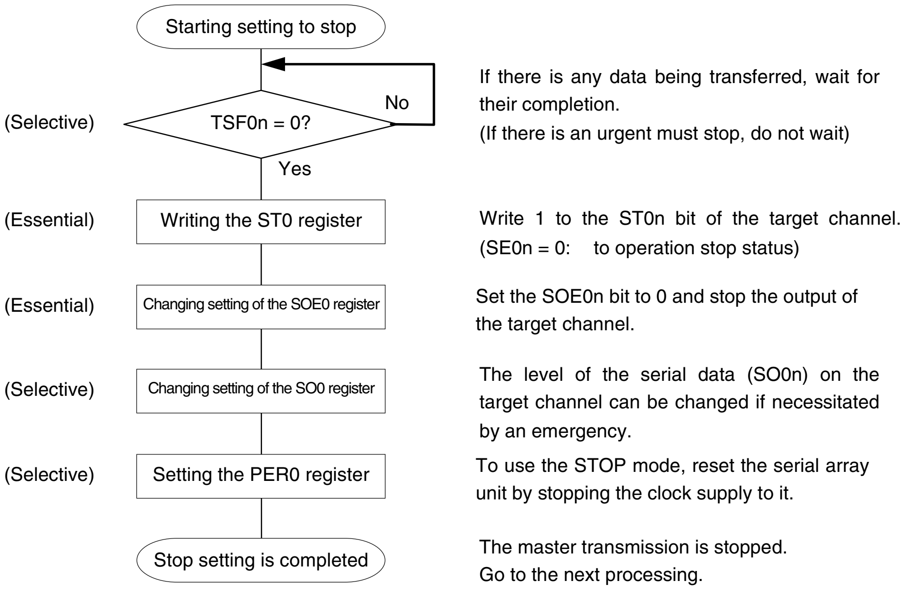
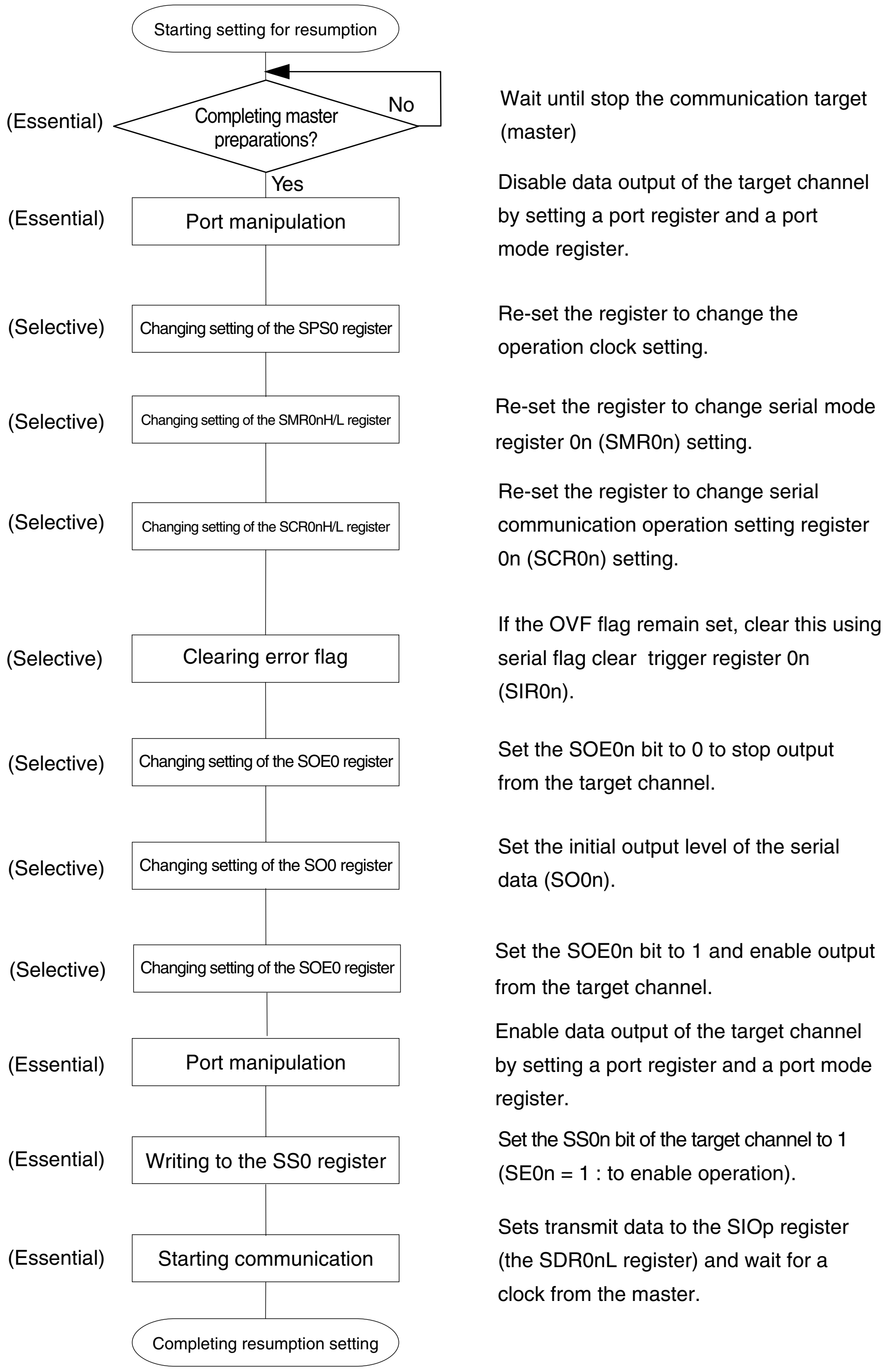


Figure 10-65. Procedure for Resuming Slave Transmission/Reception



Wait until stop the communication target (master)

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change serial mode register 0n (SMR0n) setting.

Re-set the register to change serial communication operation setting register 0n (SCR0n) setting.

If the OVF flag remain set, clear this using serial flag clear trigger register 0n (SIR0n).

Set the SOE0n bit to 0 to stop output from the target channel.

Set the initial output level of the serial data (SO0n).

Set the SOE0n 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 SS0n bit of the target channel to 1 (SE0n = 1 : to enable operation).

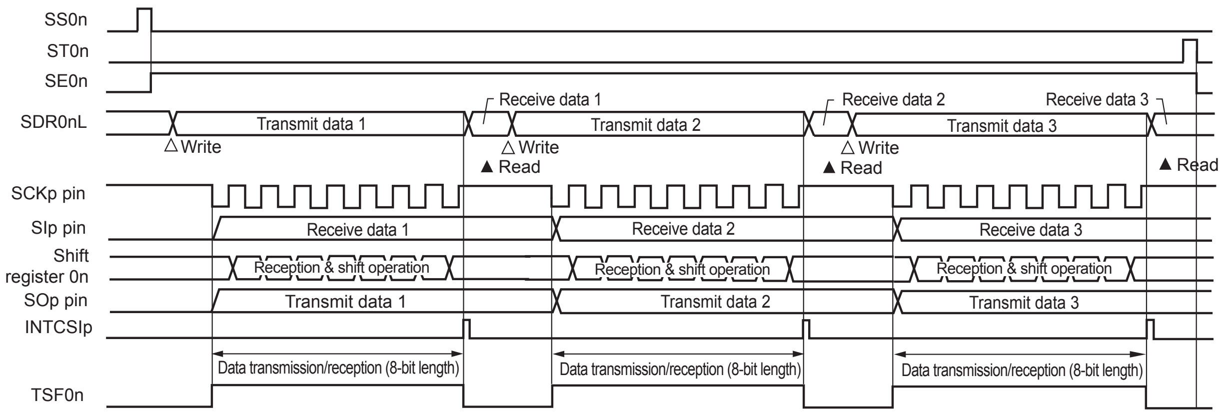
Sets transmit data to the SIOp register (the SDR0nL register) and wait for a clock from the master.

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.

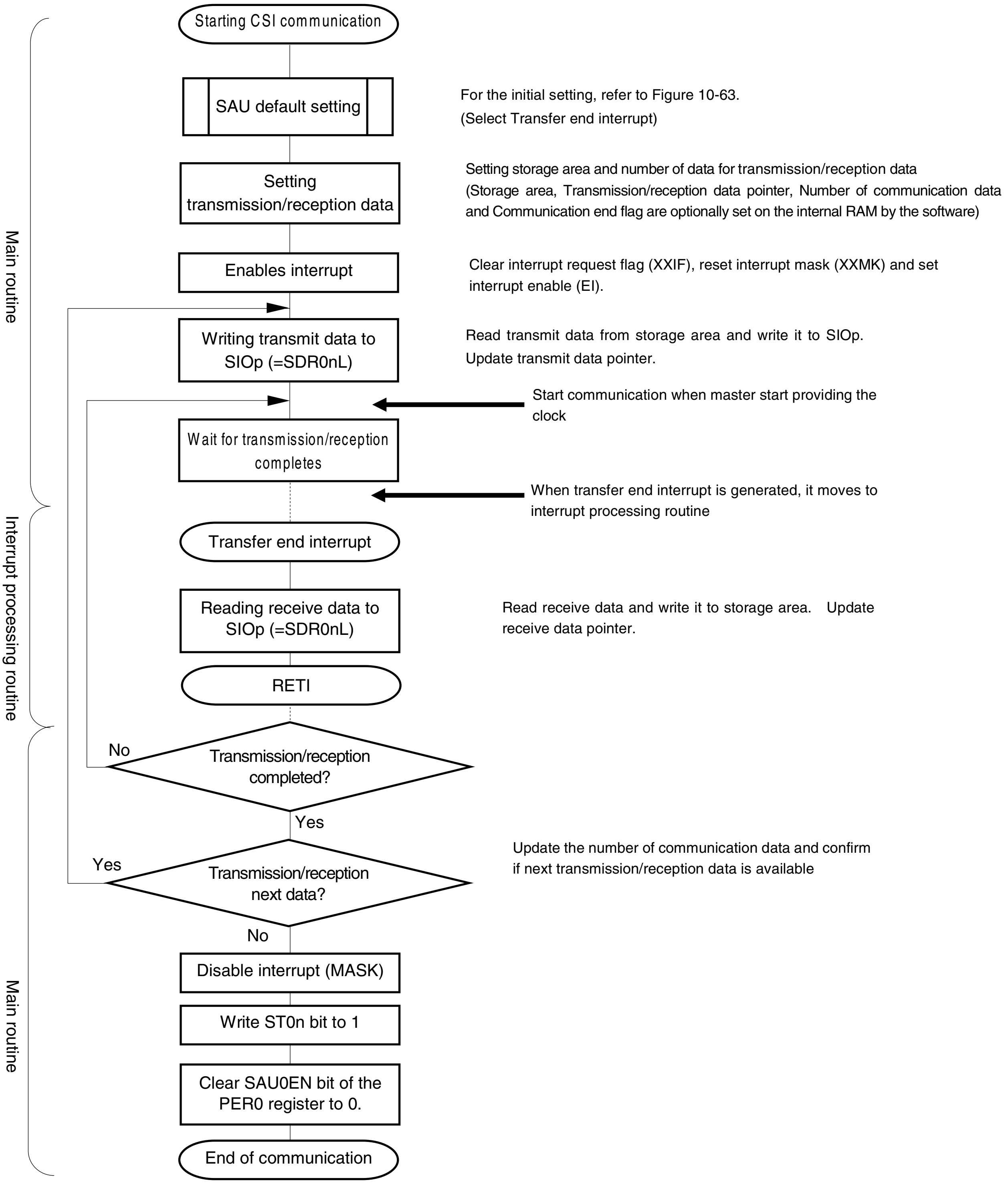
(3) Processing flow (in single-transmission/reception mode)

Figure 10-66. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAP0n = 0, CKP0n = 0)



Remark n = 0, p: CSI number (p = 00)

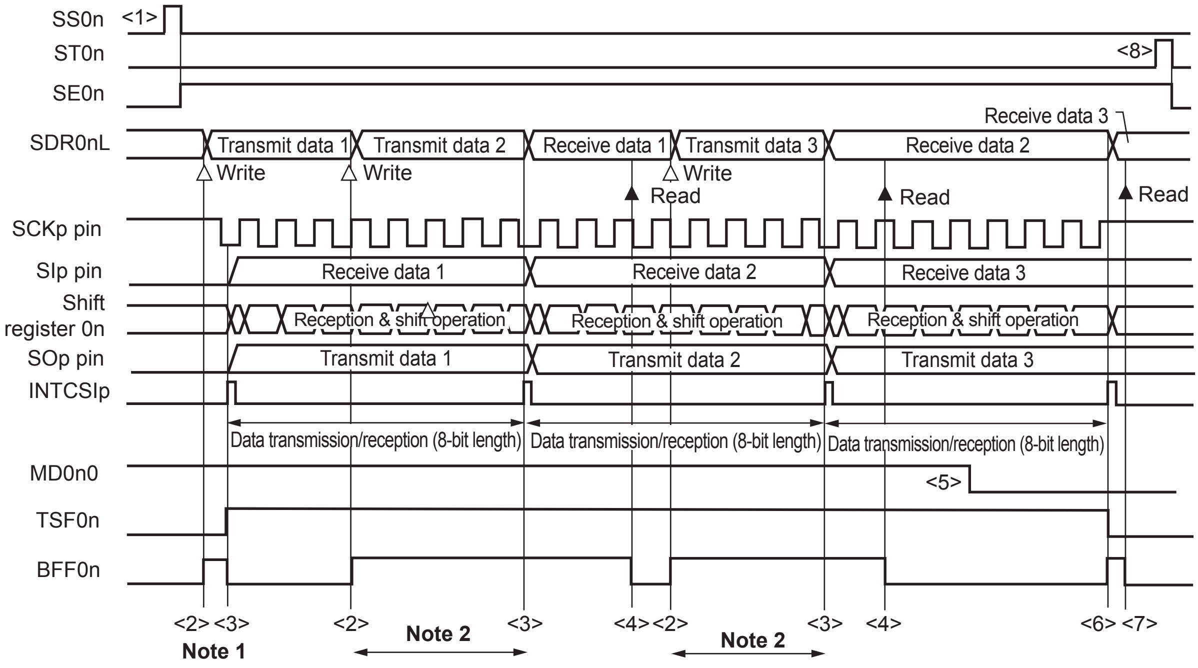
Figure 10-67. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 10-68. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAP0n = 0, CKP0n = 0)



Notes 1. If transmit data is written to the SDR0nL register while the BFF0n bit of serial status register 0n (SSR0n) is 1 (valid data is stored in serial data register 0n (SDR0nL)), the transmit data is overwritten.

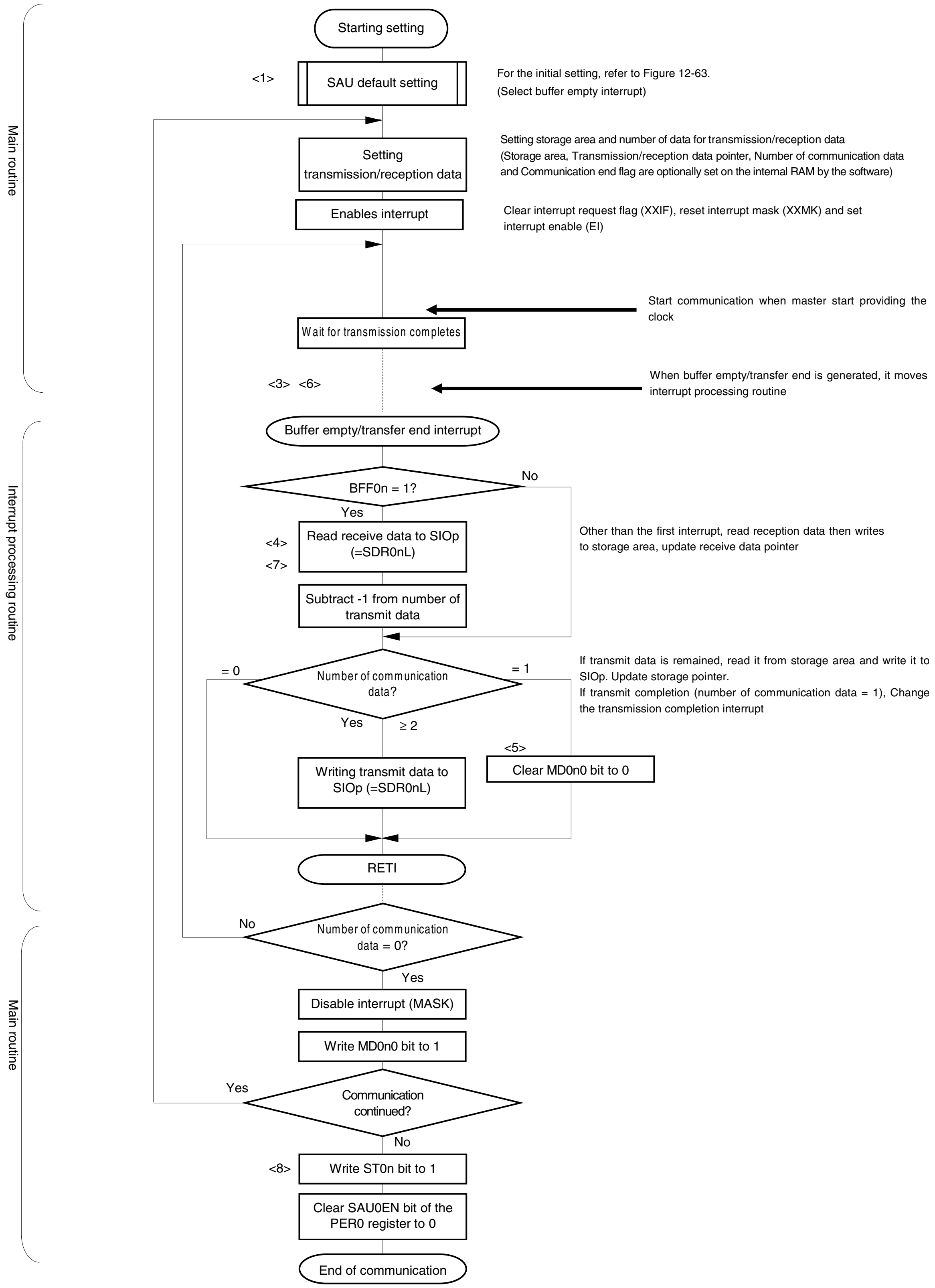
2. The transmit data can be read by reading the SDR0nL register during this period. At this time, the transfer operation is not affected.

Caution The MD0n0 bit of serial mode register 0n (SMR0nL) 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 10-69 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. n = 0, p: CSI number (p = 00)

Figure 10-69. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 10-68 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

<R> 10.5.7 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI00) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of the target channel } ÷ (SDRnH[7:1] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master}^{Note} [Hz]

Notes The permissible maximum transfer clock frequency is $f_{CLK}/6$.

Remark The value of SDR0nH[7:1] is the value of bits 7 to 1 of serial data register 0n (SDR0nH) (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (f_{MCK}) is determined by serial clock select register 0 (SPS0) and bit 7 (CKS0n) of serial mode register 0n (SMR0nH).

10.5.8 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00) communication

The procedure for processing errors that occurred during 3-wire serial I/O (CSI00) communication is described in Figure 10-70.

Figure 10-70. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register 0n (SDR0nL).	The BFF0n bit of the SSR0n 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 0n (SSR0n).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register 0n (SIR0n).	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSR0n register to the SIR0n register without modification.

Remark n = 0

10.6 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 consists 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 (an even-numbered channel) and a channel dedicated to reception (an odd-numbered channel).

[Data transmission/reception]
- Data length of 7 or 8 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse (selecting whether to reverse the level)
- Parity bit appending and parity check functions
- Stop bit appending and stop bit check functions

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]
- Framing error, parity error, or overrun error

Unit	Channel	Used as CSI	Used as UART
0	0	CSI00	UART0
	1	–	

Caution When UART operation is selected, the even-numbered channel can only be used for transmission and the odd-numbered channel can only be used for reception.

UART performs the following four types of communication operations.
- UART transmission (See **10.6.1**.)
- UART reception (See **10.6.2**.)

10.6.1 UART transmission

UART transmission is an operation to transmit data from the R7F0C80112ESP, R7F0C80212ESP to another device asynchronously (start-stop synchronization).

Of the two channels used for UART, the even-numbered channel is used for UART transmission.

UART	UART0
Target channel	Channel 0 of SAU0
Pins used	TxD0
Interrupt	INTST0
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	None
Transfer data length	7 or 8 bits (UART0 only)
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR0nH[7:1] = 2 or greater, Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]Note
Data phase	Non-inverted output (default: high level) Inverted 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 function characteristics in the electrical specifications (see **CHAPTER 21 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. n: Channel number (n = 0)

(1) Register setting

Figure 10-71. Example of Contents of Registers for UART Transmission (UART0) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol: SMR0nH

7	6	5	4	3	2	1	0
CKS0n	CCS0n						
0/1	0	0	0	0	0	0	0

Symbol: SMR0nL

7	6	5	4	3	2	1	0
						MD0n1	MD0n0
0	0	1	0	0	0	1	0/1

Operation clock (f$_{MCK}$) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol: SCR0nH

7	6	5	4	3	2	1	0
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
1	0	0	0	0	0	0/1	0/1

Symbol: SCR0nL

7	6	5	4	3	2	1	
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0/1	0/1	0	1	1	0/1

Setting of parity bit
00B: No parity
01B: Appending 0 parity
10B: Appending Even parity
11B: Appending Odd parity

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of stop bit
01B: Appending 1 bit
10B: Appending 2 bits

(c) Serial data register 0n (SDR0nH, SDR0nL)

Symbol: SDR0nH

7	6	5	4	3	2	1	0
Baud rate setting							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Transmit data setting							

TXD0

(d) Serial output level register 0 (SOL0)… Sets only the bits of the target channel.

Symbol: SOL0

7	6	5	4	3	2	1	0
							SOL00
0	0	0	0	0	0	0	0/1

0: Non-inverted transmission
1: Inverted transmission

Remarks 1. n = 0

2. ☐ Setting is fixed in the CSI master transmission mode, ▨: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 10-71. Example of Contents of Registers for UART Transmission (UART0) (2/2)

(e) Serial clock output register 0 (CKO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
CKO0	0	0	0	0	0	0	1	CKO00 ×

(f) Serial output register 0 (SO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	1	SO00 0/1 Note

0: Serial data output value is "0"
1: Serial data output value is "1"

(g) Serial output enable register 0 (SOE0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	SOE00 0/1

(h) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 ×	SS00 0/1

Note Before transmission is started, be sure to set to 1 when the SOL00 bit of the target channel is set to 0, and set to 0 when the SOL00 bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remarks 1. n = 0

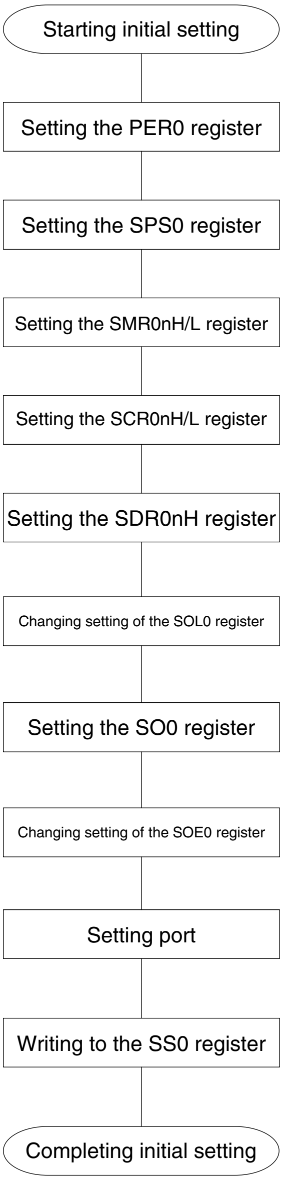
2. ▨: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 10-72. 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 (SO00).

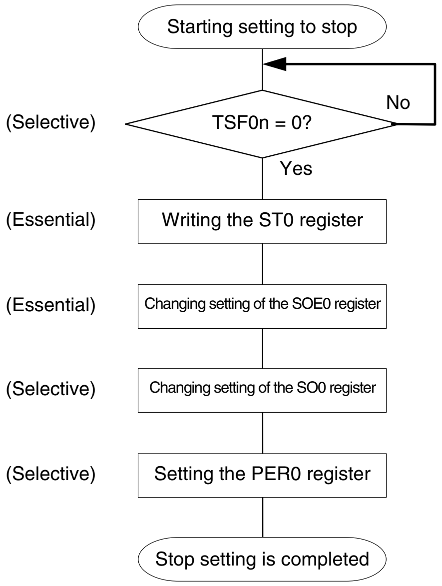
Set the SOE00 bit to 1 and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

Set the SS00 bit of the target channel to 1 and set the SE0n bit to 1 (to enable operation).

Initial setting is completed.
Set transmit data to the TXD0 register (the SDR0nL register) and start communication.

Figure 10-73. 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 ST0n bit of the target channel.
(SE0n = 0 : to operation stop status)

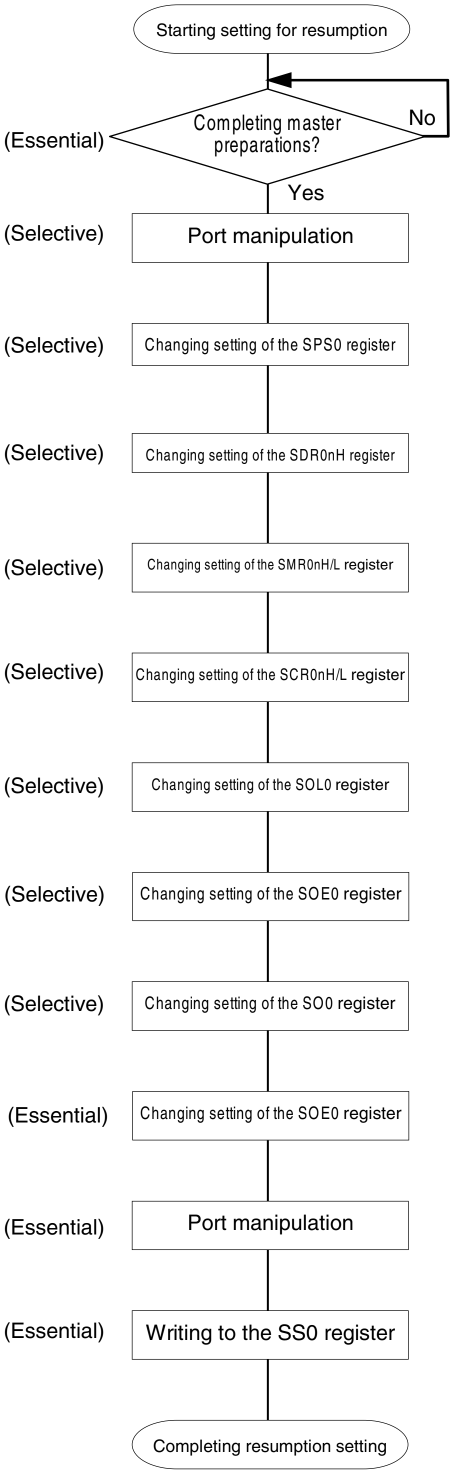
Set the SOE0n bit to 0 and stop the output of the target channel.

The level of the serial data (SO0n) 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 10-74. 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 0n (SMR0nH/L) setting.

Re-set the register to change the serial communication operation setting register 0n (SCR0n) setting.

Re-set the register to change serial output level register 0 (SOL0) setting.

Clear the SOE0n bit to 0 and stop output.

Set the initial output level of the serial data (SO0n).

Set the SOE0n bit to 1 and enable output.

Enable data output of the target channel by setting a port register and a port mode register.

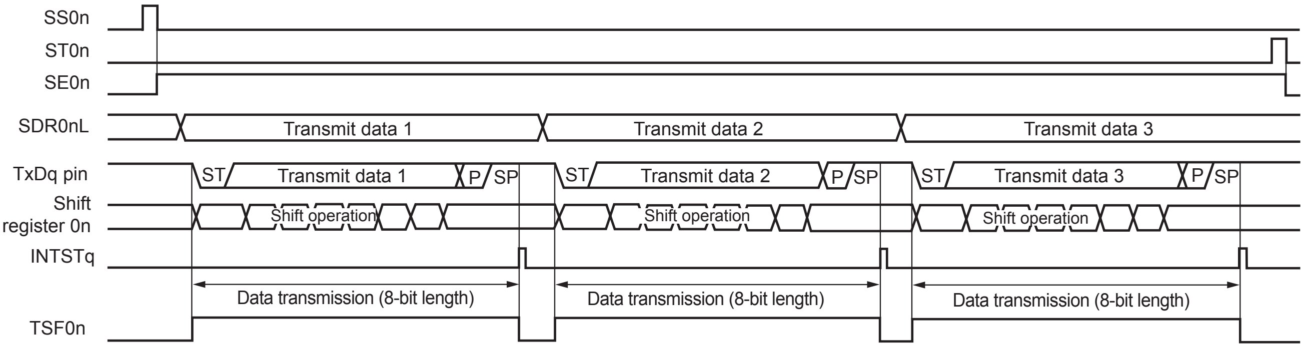
Set the SS0n bit of the target channel to 1 and set the SE0n bit to 1 (to enable operation).

Setting is completed
Sets transmit data to the TXD0 register (SDR0nL 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 stops or transmission finishes, and then perform initialization instead of restarting the transmission.

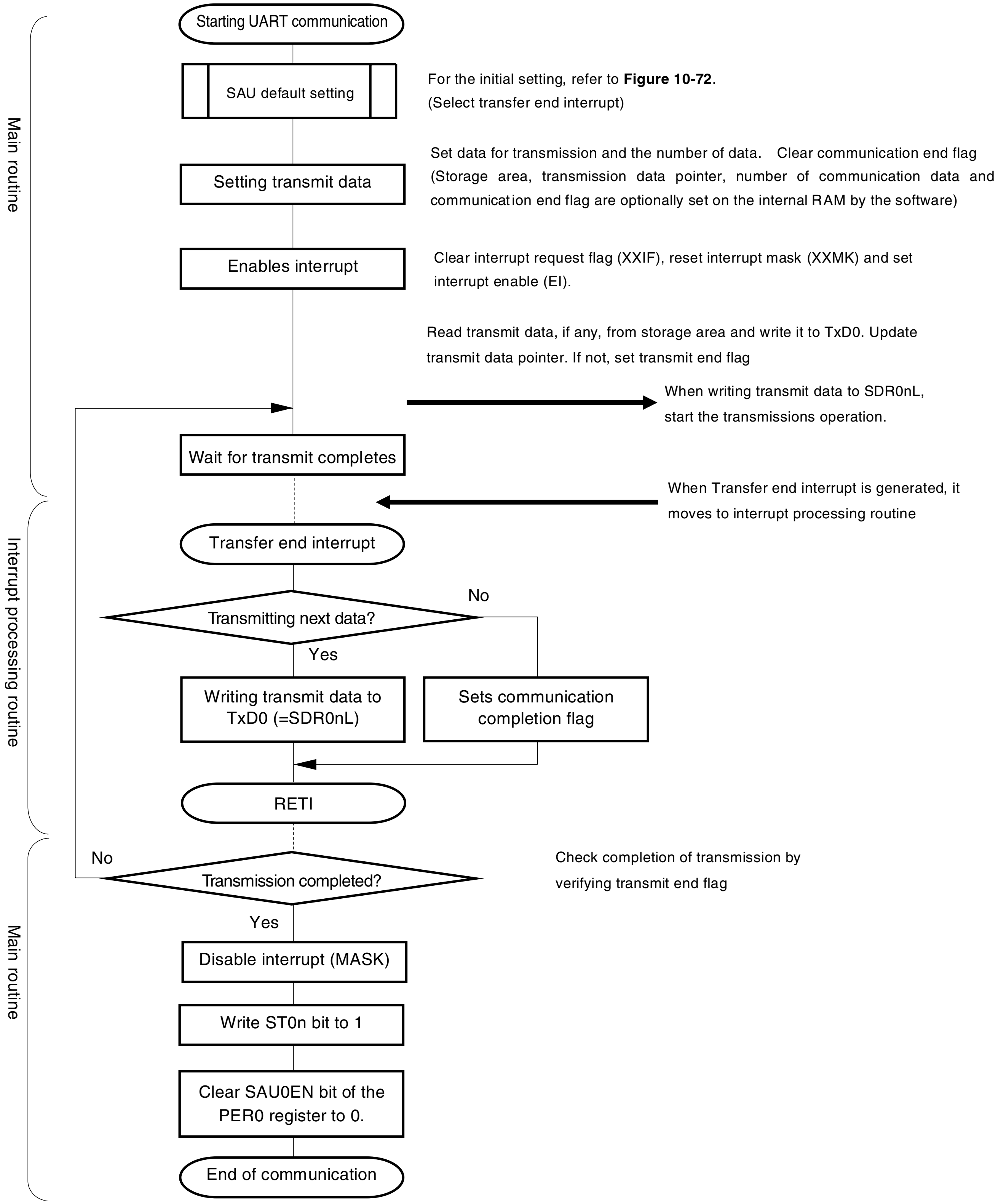
(3) Processing flow (in single-transmission mode)

Figure 10-75. Timing Chart of UART Transmission (in Single-Transmission Mode)



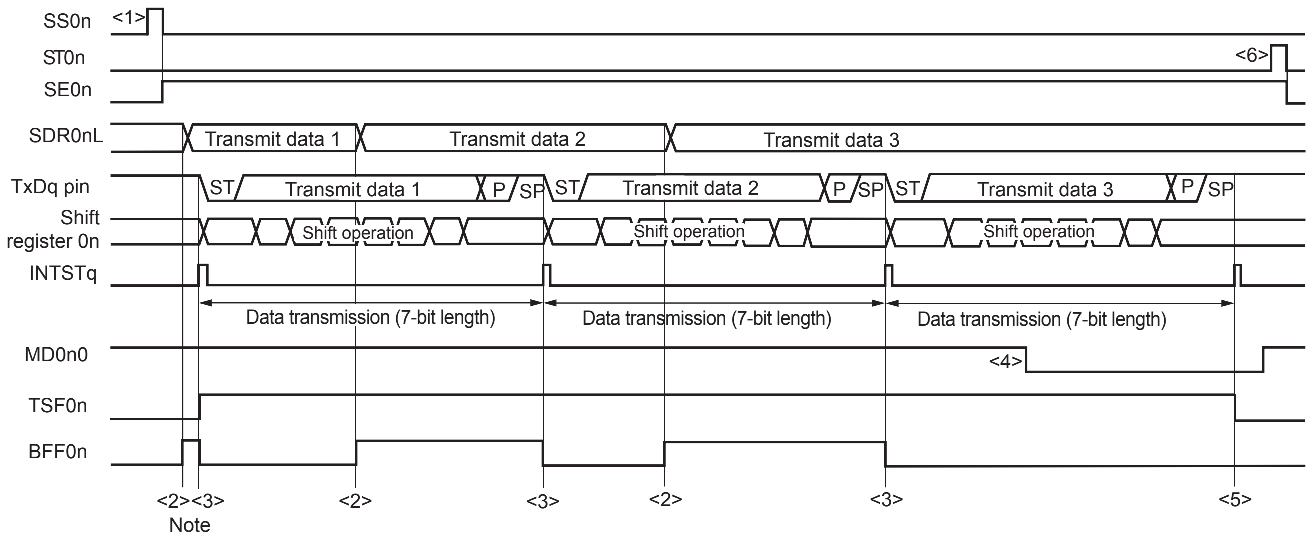
Remark q: UART number (q = 0), n = 0

Figure 10-76. Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 10-77. Timing Chart of UART Transmission (in Continuous Transmission Mode)

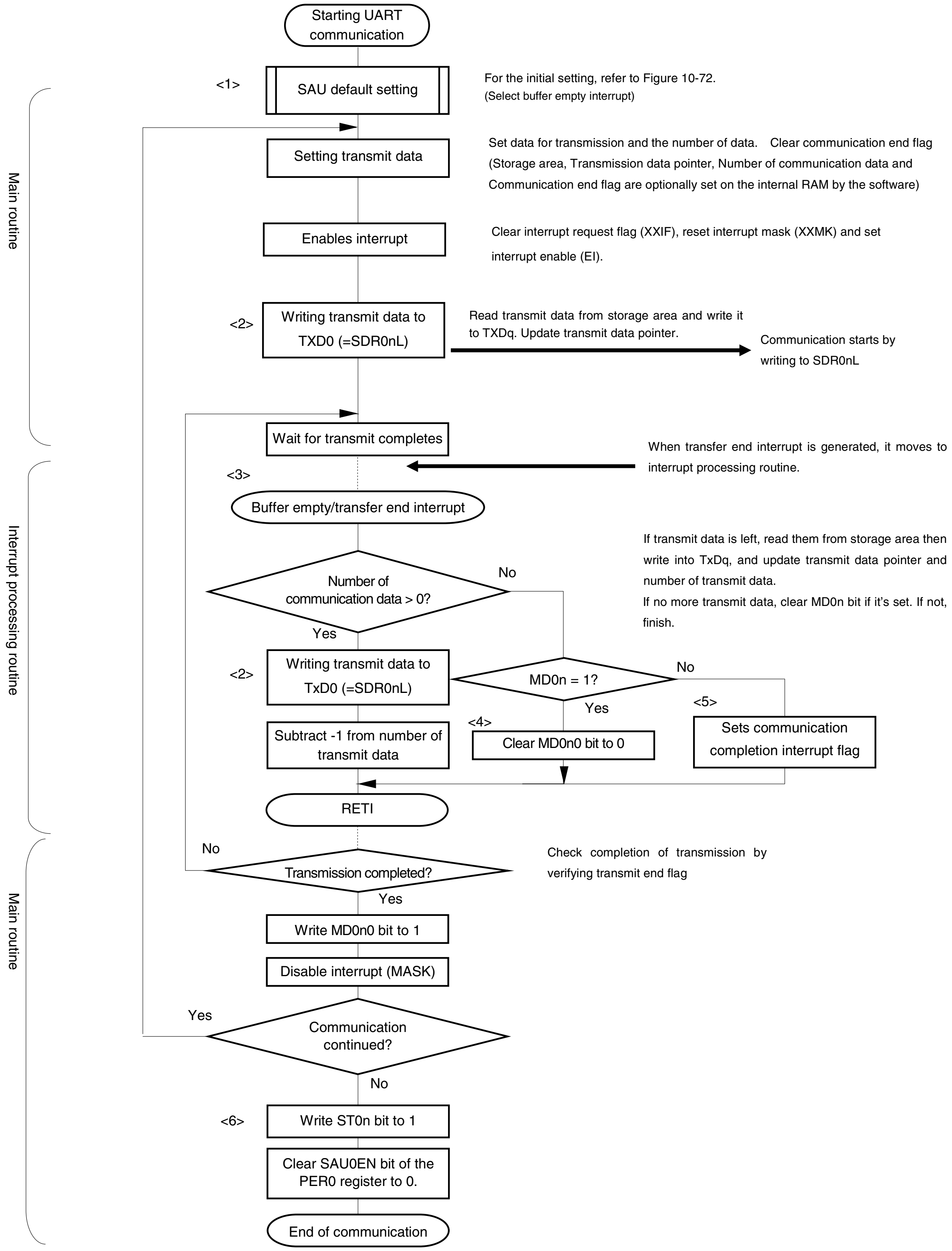


Note If transmit data is written to the SDR0nL register while the BFF0n bit of serial status register 0n (SSR0n) is 1 (valid data is stored in serial data register 0n (SDR0nL)), the transmit data is overwritten.

Caution The MD0n0 bit of serial mode register 0n (SMR0nL) 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 q: UART number (q = 0), n = 0

Figure 10-78. Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 10-77 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

10.6.2 UART reception

UART reception is an operation wherein the R7F0C80112ESP, R7F0C80212ESP 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 of SAU0
Pins used	RxD0
Interrupt	INTSR0
	Transfer end interrupt only (setting the buffer empty interrupt is prohibited)
Error interrupt	INTSRE0
Error detection flag	• Framing error detection flag (FEF0n) • Parity error detection flag (PEF0n) • Overrun error detection flag (OVF0n)
Transfer data length	7 or 8 bits (UART0 only)
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR0nH[7:1] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]^{Note}
Data phase	Non-inverted output (default: high level) Inverted output (default: low level)
Parity bit	The following selectable • No parity check • No parity specified (0 parity) • Appending even parity • Appending odd parity
Stop Bit	1 bit check
Data direction	MSB or LSB first

Note Use this operation within a range that satisfies the conditions above and the peripheral characteristics in the electrical specifications (see **CHAPTER 21 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

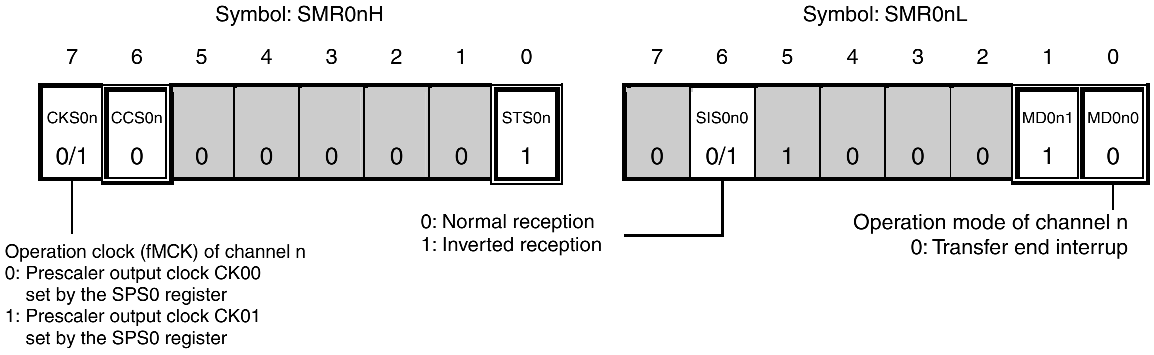
f_{CLK}: System clock frequency

2. n: Channel number (n = 1)

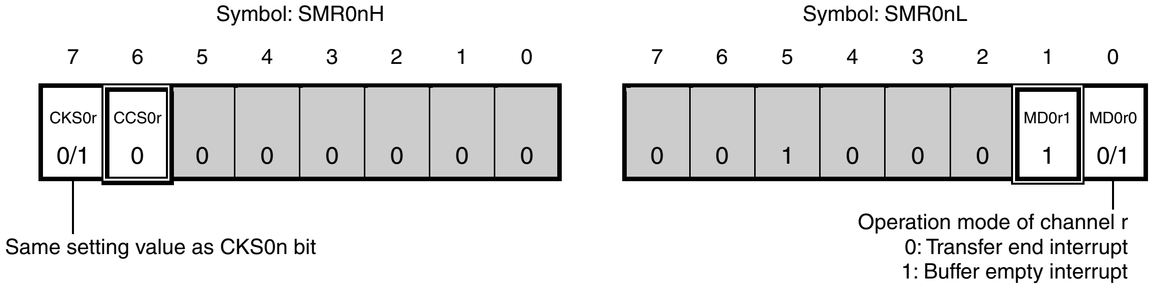
(1) Register setting

Figure 10-79. Example of Contents of Registers for UART Reception (UART0) (1/2)

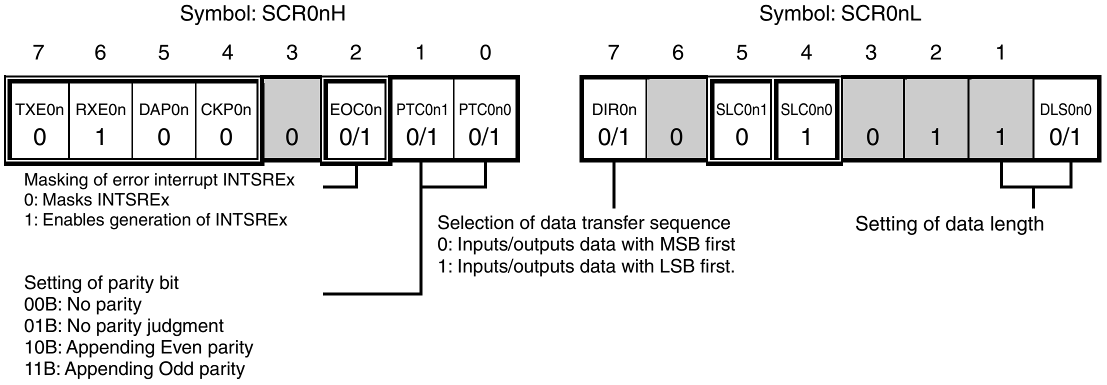
(a) Serial mode register 0n (SMR0nH, SMR0nL)



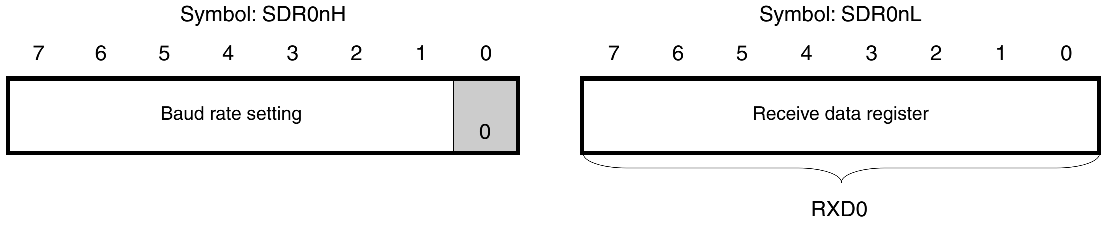
(b) Serial mode register 0r (SMR0rH, SMR0rL)



(c) Serial communication operation setting register 0n (SCR0nH, SCR0nL)



(d) Serial data register 0n (SDR0nH, SDR0nL)



Caution For UART reception, be sure to set the SMR0r register of channel r that is to be paired with channel n.

Remarks 1. n: Channel number (n = 1),
r: Channel number (r = n - 1) q: UART number (q = 0)

2. ☐: Setting is fixed in the UART master transmission mode, ▨: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 10-79. Example of Contents of Registers for UART Reception (UART0) (2/2)

(e) Serial clock output register 0 (CKO0) … The register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
CKO0	0	0	0	0	0	0	1	CKO00 ×

(f) Serial output register 0 (SO0) … The register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	1	SO00 ×

(g) Serial output enable register 0 (SOE0) … The register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	SOE00 ×

(h) Serial channel start register 0 (SS0) … Sets only the bits of the target channel is 1.

Symbol:	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0/1	SS00 ×

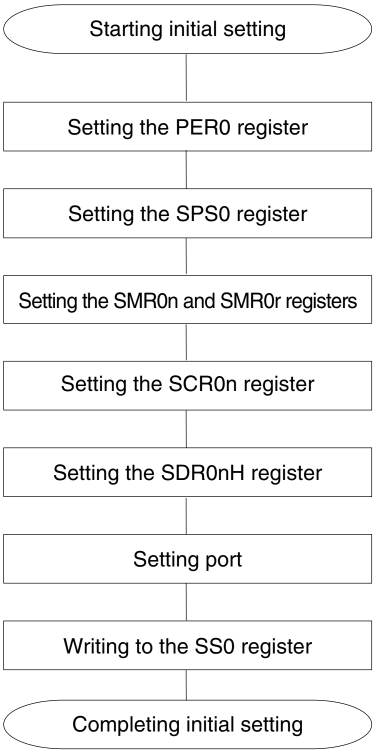
Caution For UART reception, be sure to set the SMR0r register of channel r that is to be paired with channel 0.

Remarks 1. n: Channel number (n = 1)
r: Channel number (r = n - 1) q: UART number (q = 0)

2. ▨: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 10-80. 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.

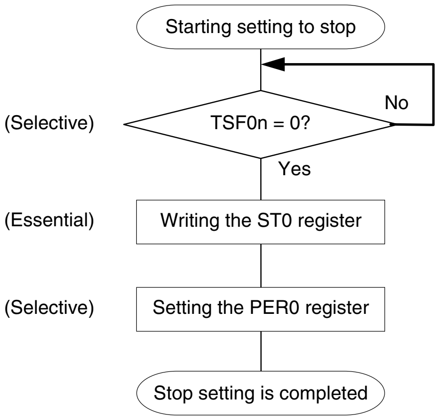
Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Enable data input of the target channel by setting a port register and a port mode register.

Set the SS0n bit of the target channel to 1 and set the SE0n bit to 1 (to enable operation). Become wait for start bit detection.

Caution After setting the RXE0n bit of SCR0n register to 1, be sure to set SS0n to 1 after 4 or more f_{CLK} clocks have elapsed.

Figure 10-81. Procedure for Stopping UART Reception



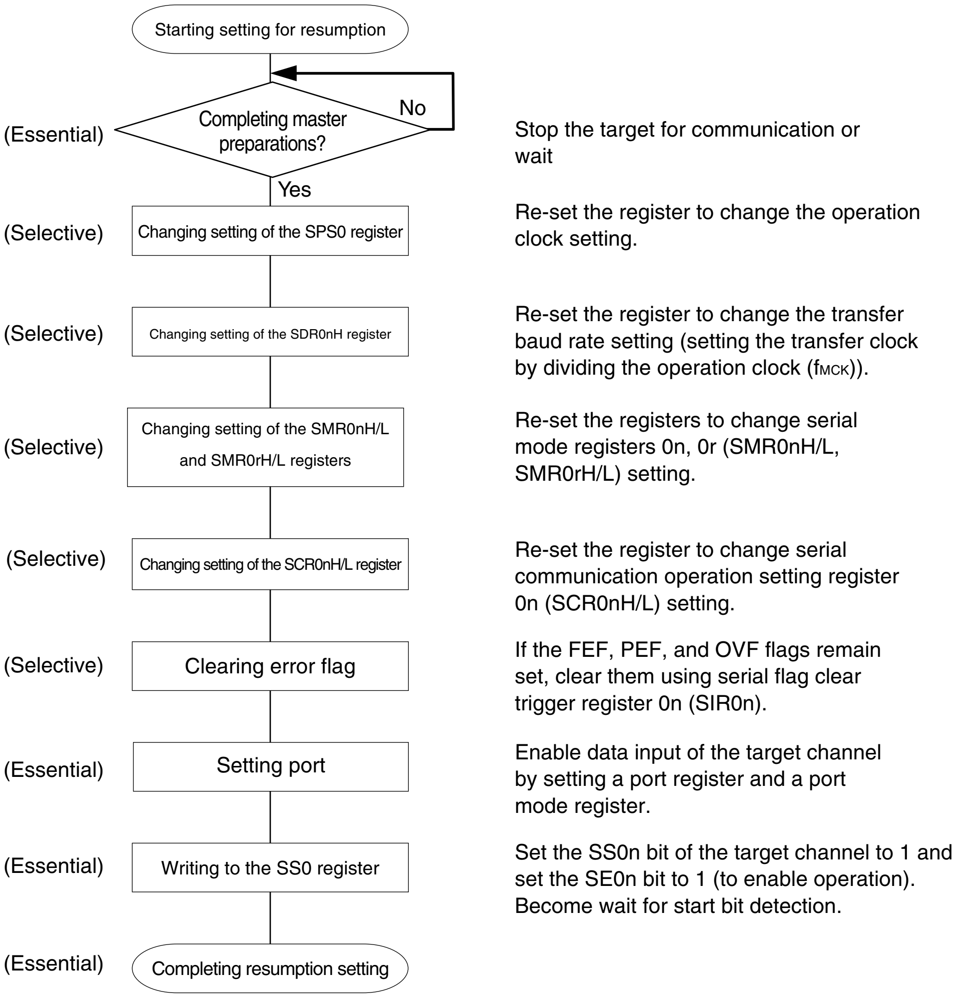
If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait)

Write 1 to the ST0n bit of the target channel.
(SE0n = 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 10-82. Procedure for Resuming UART Reception



Stop the target for communication or wait

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the registers to change serial mode registers 0n, 0r (SMR0nH/L, SMR0rH/L) setting.

Re-set the register to change serial communication operation setting register 0n (SCR0nH/L) setting.

If the FEF, PEF, and OVF flags remain set, clear them using serial flag clear trigger register 0n (SIR0n).

Enable data input of the target channel by setting a port register and a port mode register.

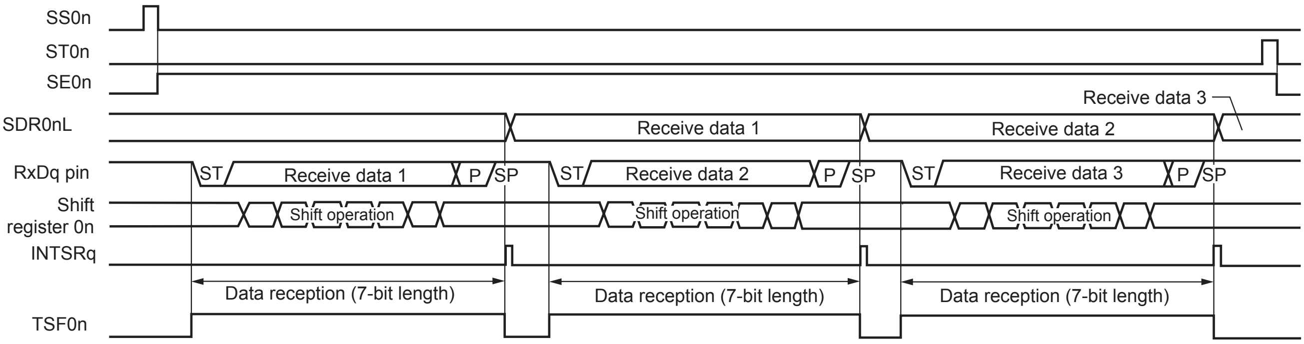
Set the SS0n bit of the target channel to 1 and set the SE0n bit to 1 (to enable operation). Become wait for start bit detection.

Caution **After setting the RXE0n bit of SCR0n register to 1, be sure to set SS0n to 1 after 4 or more f_{CLK} clocks have elapsed.**

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

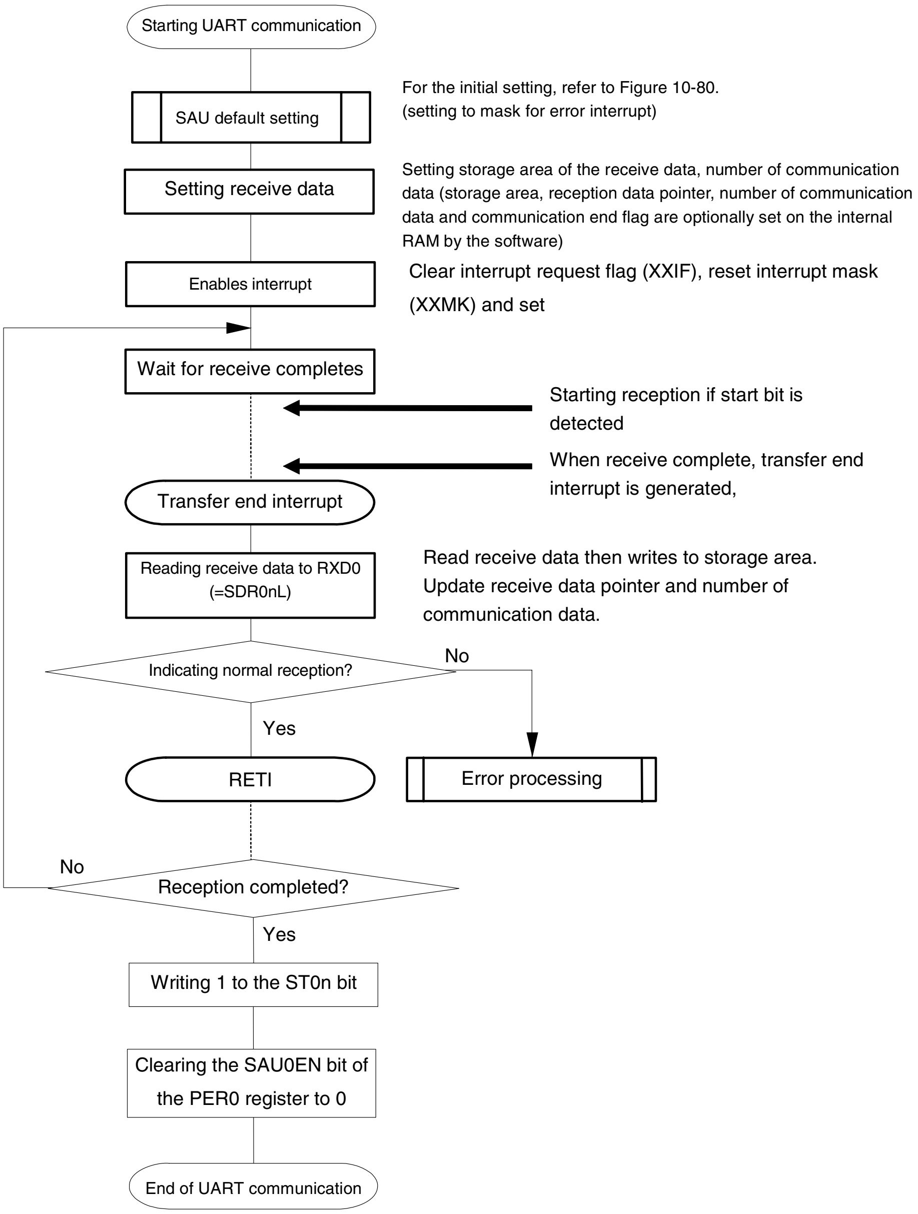
(3) Processing flow

Figure 10-83. UART Reception Timing Chart



Remark n: Channel number (n = 1)
r: Channel number (r = n - 1) q: UART number (q = 0)

Figure 10-84. Flowchart of UART Reception



For the initial setting, refer to Figure 10-80.
(setting to mask for error 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

Starting reception if start bit is detected

When receive complete, transfer end interrupt is generated,

Read receive data then writes to storage area. Update receive data pointer and number of communication data.

10.6.3 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} ÷ (SDR0nH[7:1] + 1) ÷ 2 [bps]

Caution Setting serial data register 0n (SDR0nH) SDR0nH[7:1] = (0000000B to 0000001B) is prohibited.

Remarks 1. When UART is used, the value of SDR0nH[7:1] is the value of bits 15 to 9 of the SDR0nH register (0000010B to 1111111B) and therefore is 2 to 127.

2. n = 0, 1

The operation clock (f_{MCK}) is determined by serial clock select register 0 (SPS0) and bit 7 (CKS0n) of serial mode register 0n (SMR0nH).

(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} = 20 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 20 MHz			
	Operation Clock (f_{MCK})	SDR0nH[7:1]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	64	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^8$	64	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^7$	64	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^6$	64	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^5$	64	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^4$	64	9615.38 bps	+0.16 %
19200 bps	$f_{CLK}/2^3$	64	19230.8 bps	+0.16 %
31250 bps	$f_{CLK}/2^3$	39	31250.0 bps	±0.0 %
38400 bps	$f_{CLK}/2^2$	64	38461.5 bps	+0.16 %
76800 bps	$f_{CLK}/2$	64	76923.1 bps	+0.16 %
153600 bps	f_{CLK}	64	153846 bps	+0.16 %
312500 bps	f_{CLK}	31	312500 bps	±0.0 %

(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.

$$(\text{Maximum receivable baud rate}) = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$(\text{Minimum receivable baud rate}) = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **10.6.3 (1) Baud rate calculation expression**.)

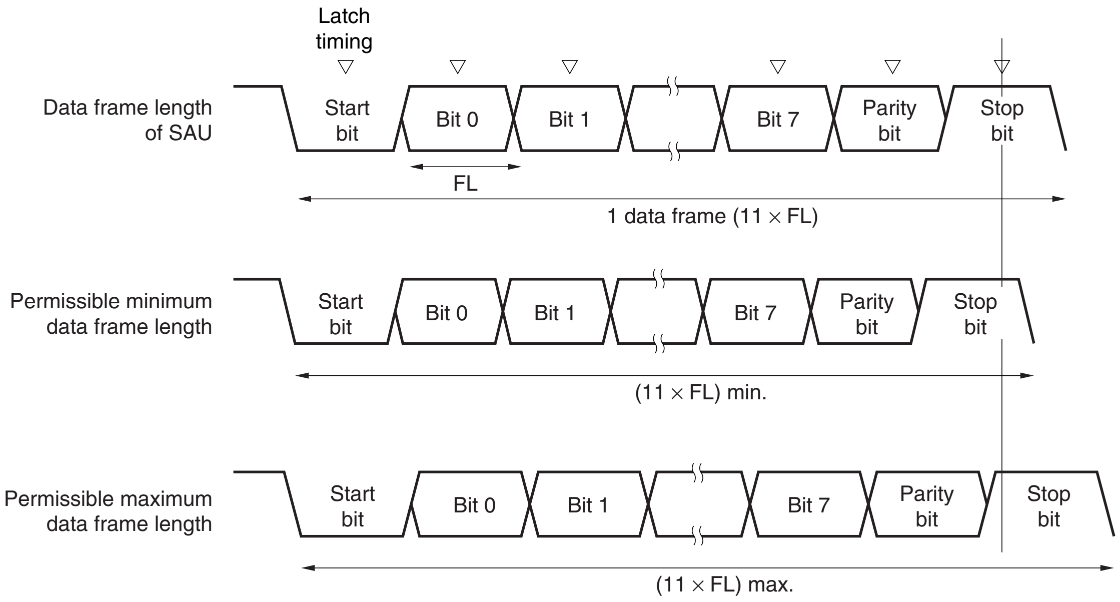
k: SDR0nH[7:1] + 1

Nfr: 1 data frame length [bits]

= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark n = 1

Figure 10-85. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 10-84 the timing of latching receive data is determined by the division ratio set by bits 7 to 1 of serial data register 0nH (SDR0nH) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

10.6.4 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 10-86 and 10-87.

Figure 10-86. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register 0n (SDR0nL).	The BFF0n bit of the SSR0n 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 0n (SSR0n).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register 0n (SIR0n).	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSR0n register to the SIR0n register without modification.

Figure 10-87. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register 0n (SDR0nL).	The BFF0n bit of the SSR0n 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 0n (SSR0n).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register 0n (SIR0n).	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSR0n register to the SIR0n register without modification.
Sets the ST0n bit of serial channel stop register 0 (ST0) to 1.	The SE0n bit of serial channel enable status register 0 (SE0) 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 SS0n bit of serial channel start register 0 (SS0) to 1.	The SE0n bit of serial channel enable status register 0 (SE0) is set to 1 and channel n is enabled to operate.	

Remark n = 0, 1

CHAPTER 11 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.

		Number of interrupt sources
Maskable interrupts	External	3
	Internal	8

11.1 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, PR10L, PR10H).

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 11-1**.

A standby release signal is generated and STOP and HALT 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.

11.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to four reset sources (see **Table 11-1**). 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 11-1. Interrupt Source List

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	0	INTWDTI	Watchdog timer interval (75% of overflow time +3/(4 x f_{IL}))	Internal	0004H	(a)
	1	INTP0	Pin input edge detection	External	0006H	(b)
	2	INTP1			0008H	
	3	INTST0/ INTCSI00/	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt	Internal	000AH	(a)
	4	INTSR0	UART0 reception transfer end		000CH	
	5	INTSRE0	UART0 reception communication error occurrence		000EH	
	6	INTTM01H	End of counting or start of operations by timer channel 1 (at higher 8-bit timer operation)		0010H	
	7	INTTM00	End of counting or start of operations by timer channel 0		0012H	
	8	INTTM01	End of counting, completion of capture, or start of operations by timer channel 1 (at 16-bit or lower 8-bit timer operation)		0014H	
	9	INTAD	End of A/D conversion		0016H	
	10	INTKR	Key return signal detection	External	0018H	(c)
Software	–	BRK	Execution of BRK instruction	–	007EH	(d)
Reset	–	RESET	$\overline{\text{RESET}}$ pin input	–	0000H	–
		SPOR	Selectable power-on-reset			
		WDT	Overflow of watchdog timer			
		TRAP	Execution of illegal instruction[Note 3]			

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 10 indicates the lowest priority.

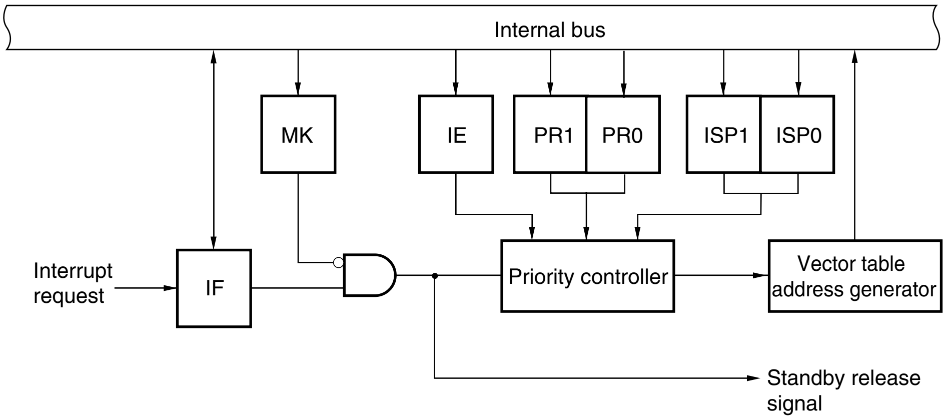
2. Basic configuration types (a) to (d) correspond to (a) to (d) in Figure 11-1.

3. When the instruction code in FFH is executed.

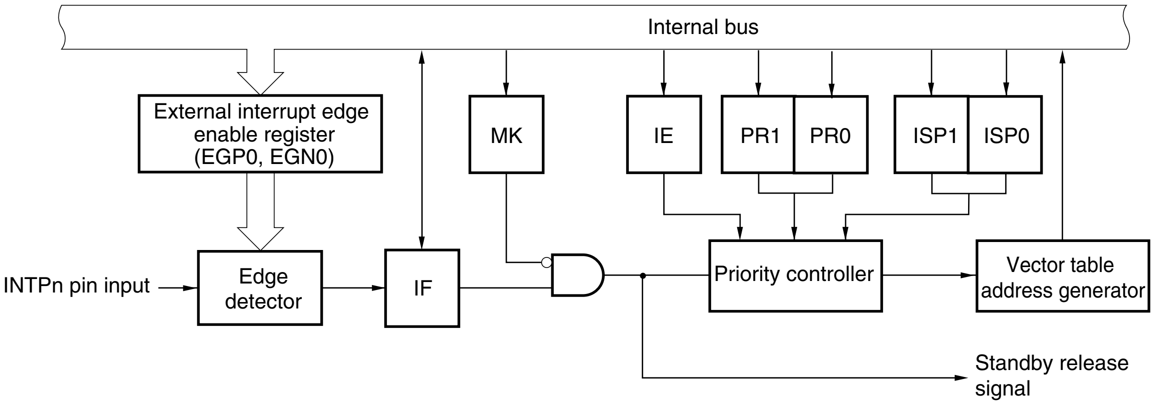
No reset is issued even if an illegal instruction is executed during emulation with the on-chip debug emulator.

Figure 11-1. Basic Configuration of Interrupt Function (1/2)

(a) Internal maskable interrupt



(b) External maskable interrupt (INTPn)

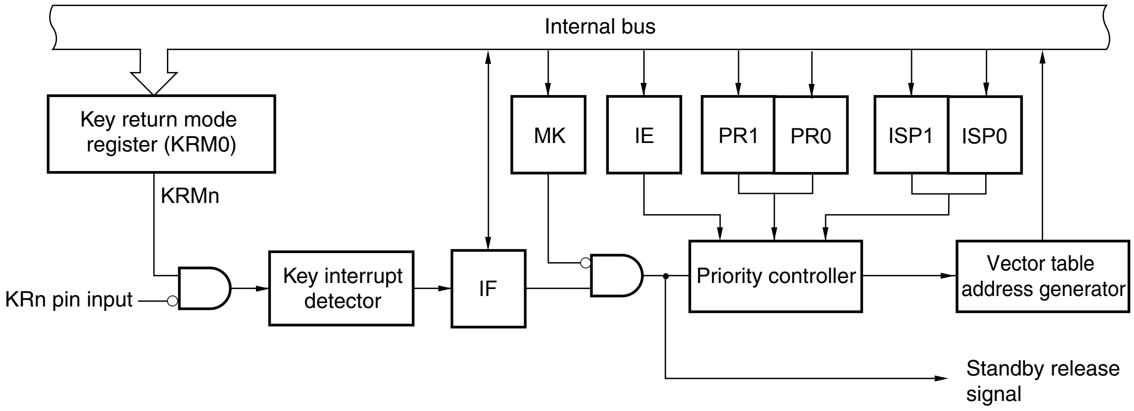


IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP0:	In-service priority flag 0
ISP1:	In-service priority flag 1
MK:	Interrupt mask flag
PR0:	Priority specification flag 0
PR1:	Priority specification flag 1

Remark n = 0, 1

Figure 11-1. Basic Configuration of Interrupt Function (2/2)

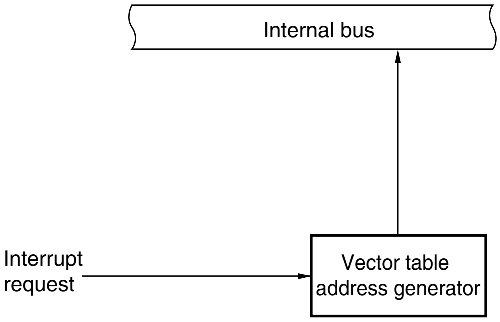
(c) External maskable interrupt (INTKR)



IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

Remark n = 0 to 5

(d) Software interrupt



11.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H)
- Interrupt mask flag registers (MK0L, MK0H)
- Priority specification flag registers (PR00L, PR00H, PR10L, PR10H)
- External interrupt rising edge enable register (EGP0)
- External interrupt falling edge enable register (EGN0)
- Program status word (PSW)

Table 11-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 11-2. Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L
INTP0	PIF0		PMK0		PPR00, PPR10	
INTP1	PIF1		PMK1		PPR01, PPR11	
INTST0[Note] INTCSI00[Note]	STIF0[Note] CSIIF00[Note]		STMK0[Note] CSIMK00[Note]		STPR00, STPR10[Note] CSIPR000, CSIPR100[Note]	
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10	
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10	
INTTM01H	TMIF01H		TMMK01H		TMPR001H, TMPR101H	
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTTM01	TMIF01	IF0H	TMMK01	MK0H	TMPR001, TMPR101	PR00H PR10H
INTAD	ADIF		ADMK		ADPR0, ADPR1	
INTKR	KRIF		KRMK		KRPR0, KRPR1	

Note If interrupt source INTST0, or INTCSI00 occurs, bit 3 of the IF0L register is set to 1.
Bit 3 of the MK0L, and PR00L and PR10L registers supports these two interrupt sources.

11.3.1 Interrupt request flag registers (IF0L, IF0H)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when the interrupt request is acknowledged, a reset signal is generated, or an instruction is executed.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

IF0L, and IF0H registers can be set by a 1-bit or 8-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 11-2. Format of Interrupt Request Flag Registers (IF0L, IF0H)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	TMIF00	TMIF01H	SREIF0	SRIF0	STIF0 CSIIF00	PIF1	PIF0	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
IF0H	0	0	0	0	0	KRIF	ADIF	TMIF01

XXIFXX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. Do not change undefined bit data.

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 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.

11.3.2 Interrupt mask flag registers (MK0L, MK0H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

The MK0L, and MK0H registers can be set by a 1-bit or 8-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 11-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	TMMK00	TMMK01H	SREMK0	SRMK0	STMK0 CSIMK00	PMK1	PMK0	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
MK0H	1	1	1	1	1	KRMK	ADMK	TMMK01

XXMKXX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Do not change undefined bit data.

11.3.3 Priority specification flag registers (PR00L, PR00H, PR10L, PR10H)

The priority specification flag registers are used to set the priority level of the corresponding maskable interrupt.

A priority level is set by using the PR0xy and PR1xy registers in combination (0xy = 0L, 0H).

The PR00L, PR00H, PR01L, PR10L, and PR10H registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 11-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR10L, PR10H)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	TMPR000	TMPR001H	SREPR00	SRPR00	STPR00 CSIPR000	PPR01	PPR00	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	TMPR100	TMPR101H	SREPR10	SRPR10	STPR10 CSIPR100	PPR11	PPR10	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR00H	1	1	1	1	1	KRPR0	ADPR0	TMPR001

Address: FFFEDH After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR10H	1	1	1	1	1	KRPR1	ADPR1	TMPR101

XXPR1X	XXPR0X	Priority Level Selection
0	0	Specifying level 0 (high priority)
0	1	Specifying level 1
1	0	Specifying level 2
1	1	Specifying level 3 (low priority)

Caution Do not change undefined bit data.

11.3.4 External interrupt rising edge enable register (EGP0), external interrupt falling edge enable register (EGN0)

These registers specify the valid edge for INTP0, and INTP1.

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 11-5. Format of External Interrupt Rising Edge Enable Register (EGP0) and External Interrupt Falling Edge Enable Register (EGN0)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	0	0	0	0	0	0	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	0	0	0	0	0	0	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0, 1)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

<R> **Caution When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge. When switching the input port pins to the output mode, set the port mode register (PMxx) to 0 after disabling the edge detection (by setting EGPn and EGNn to 0).**

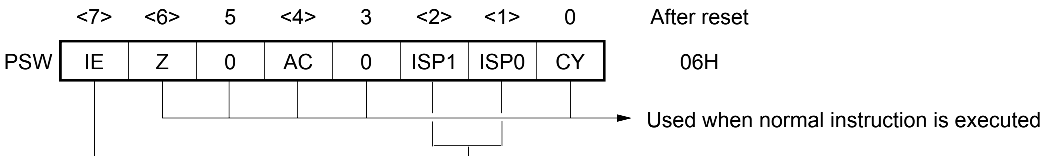
11.3.5 Program status word (PSW)

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. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are 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 11-6. 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

11.4 Interrupt Servicing Operations

11.4.1 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 servicing is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority vectored 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 11-3 below.

For the interrupt request acknowledgment timing, see **Figures 11-8** and **11-9**.

Table 11-3. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
Servicing time	11 clocks	18 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.

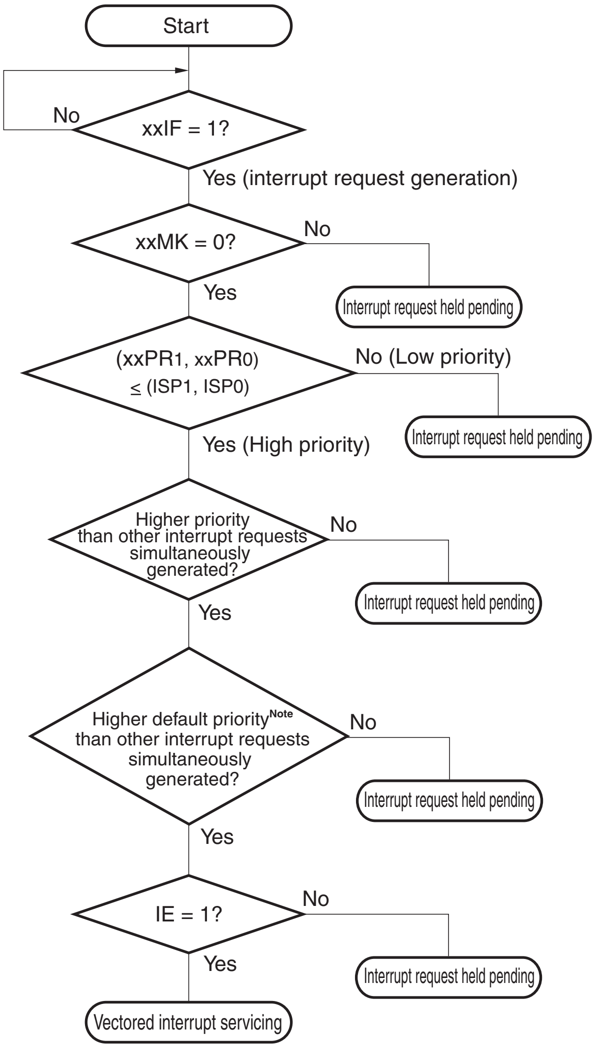
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 11-7 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.

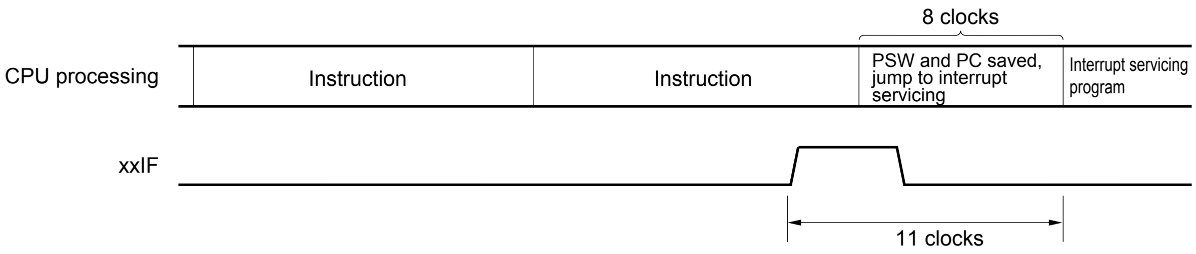
Figure 11-7. Interrupt Request Acknowledgment Processing Algorithm



××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 11-6**)

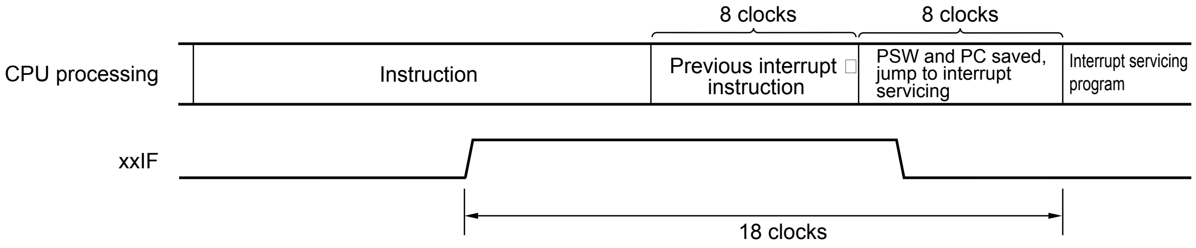
Note For the default priority, refer to **Table 11-1 Interrupt Source List**.

Figure 11-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/fCLK (fCLK: CPU clock)

Figure 11-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/fCLK (fCLK: CPU clock)

11.4.2 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.

11.4.3 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 11-4 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 11-10 shows multiple interrupt servicing examples.

Table 11-4. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0 ISP0 = 0	○	×	×	×	×	×	×	×	○
	ISP1 = 0 ISP0 = 1	○	×	○	×	×	×	×	×	○
	ISP1 = 1 ISP0 = 0	○	×	○	×	○	×	×	×	○
	ISP1 = 1 ISP0 = 1	○	×	○	×	○	×	○	×	○
Software interrupt		○	×	○	×	○	×	○	×	○

Remarks 1. ○: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

3. ISP0, ISP1, and IE are flags contained in the PSW.

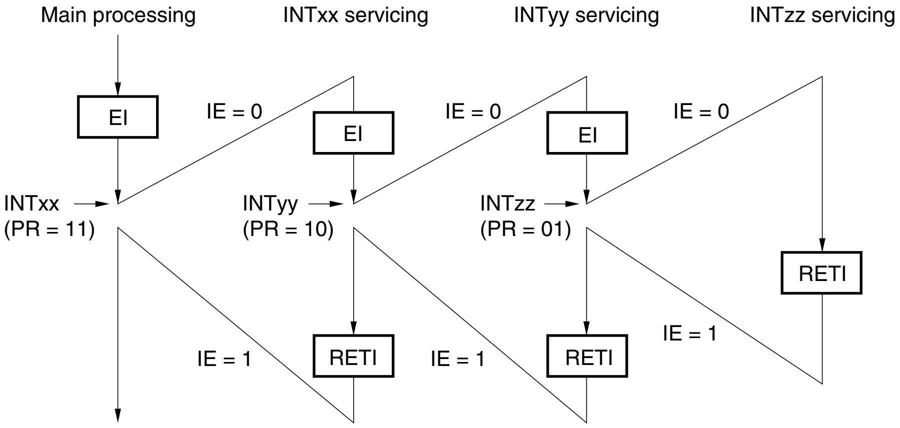
ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.
ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.
ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.
ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment.
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

4. PR is a flag contained in the PR00L, PR00H, PR10L, PR10H 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)

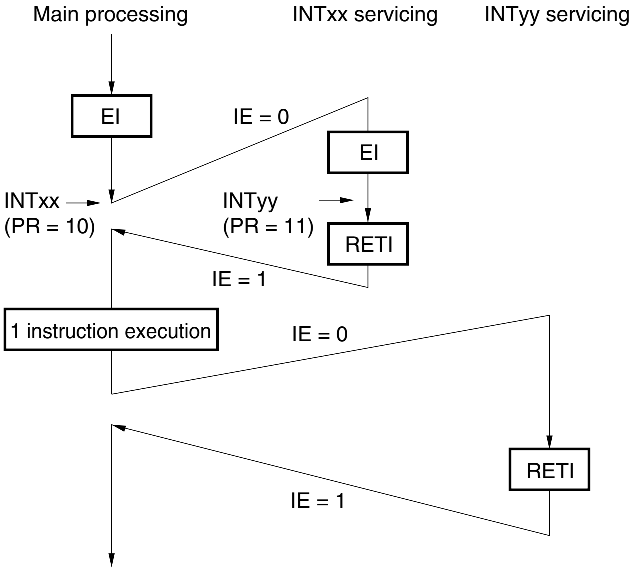
Figure 11-10. Examples of Multiple Interrupt Servicing (1/2)

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

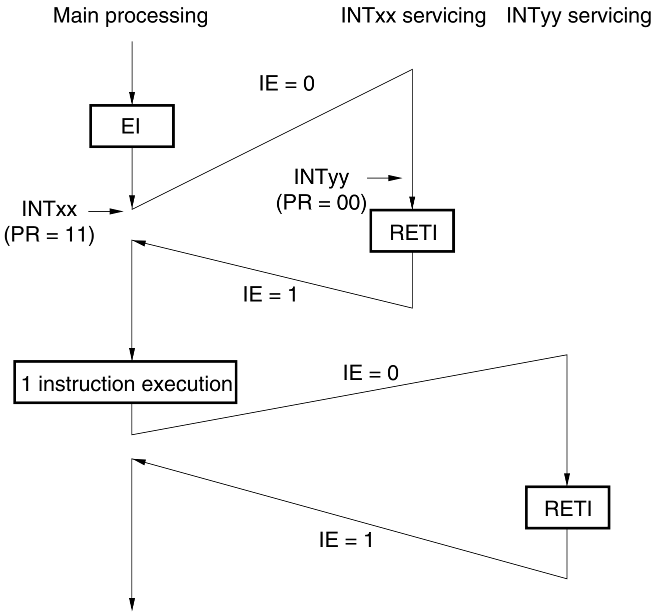


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

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 11-10. 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.

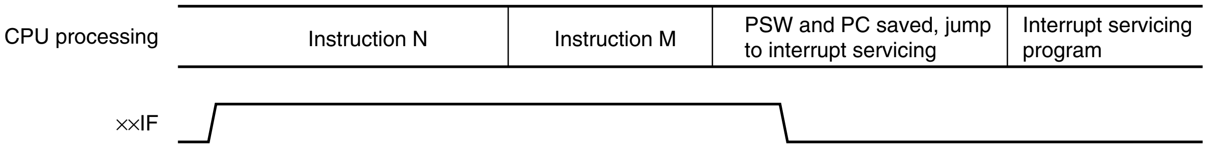
11.4.4 Interrupt request hold

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
- Instructions that write data for the IF0L, IF0H, MK0L, MK0H, PR00L, PR00H, PR10L, and PR10H registers

Figure 11-11 shows the timing at which interrupt requests are held pending.

Figure 11-11. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 12 KEY INTERRUPT FUNCTION

12.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by setting the key return mode register (KRM) and inputting a rising edge/falling edge to the key interrupt input pins (KR0 to KR5).

<R>

Table 12-1. Assignment of Key Interrupt Detection Pins

Key Interrupt Pins	Key return mode registers (KRM0)	Key return flag register (KRF)
KR0	KRM00	KRF0
KR1	KRM01	KRF1
KR2	KRM02	KRF2
KR3	KRM03	KRF3
KR4	KRM04	KRF4
KR5	KRM05	KRF5

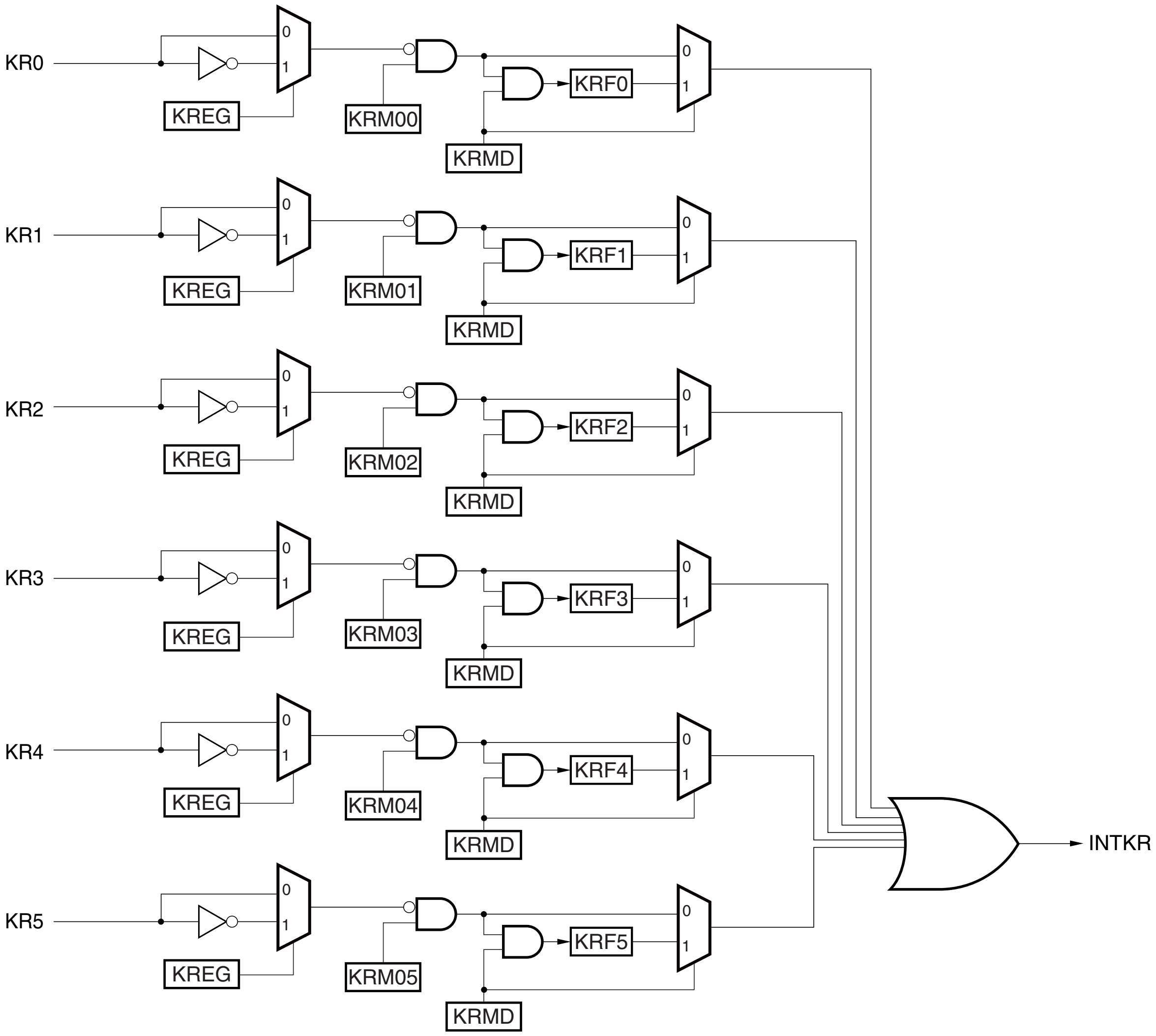
12.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 12-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return control register (KRCTL) Key return mode register (KRM0) Key return flag register (KRF) Port mode registers 0 and 4 (PM0, PM4) Port registers 0, 4, and 12 (P0, P4, P12)

Figure 12-1. Block Diagram of Key Interrupt



12.3 Register Controlling Key Interrupt

The key interrupt function is controlled by the following five registers:

- Key return control register (KRCTL)
- Key return mode register (KRM0)
- Key return flag register (KRF)
- Port mode registers 0 and 4 (PM0, PM4)
- Port registers 0, 4, and 12 (P0, P4, P12)

12.3.1 Key return control register (KRCTL)

This register controls the usage of the key return flags (KRF0 to KRF5) 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 12-2. Format of Key Return Control Register (KRCTL)

Address: FFF34H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRCTL	KRMD	0	0	0	0	0	0	KREG

KRMD	Usage of key return flags (KRF0 to KRF5)
0	Does not use key return flags
1	Uses key return flags

KREG	Selection of detection edge (KR0 to KR5)
0	Falling edge
1	Rising edge

KREG	KREG	Interrupt function
0	0	Key interrupt, external interrupt (specified by the port level) Note
0	1	External interrupt (specified by the port level)
1	0	External interrupt (specified by the flag)
1	1	

Note When the falling edge is detected, the function and operation of the external interrupt are the same as those of the key interrupt.

12.3.2 Key return mode register (KRM0)

This register sets the key interrupt mode.

The KRM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 12-3. Format of Key Return Mode Register (KRM0)

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM0	0	0	KRM05	KRM04	KRM03	KRM02	KRM01	KRM00

KRM0n	Key interrupt mode control (n = 0 to 5)
0	Does not detect key interrupt signal
1	Detects key interrupt signal

<R> **Cautions 1. When a key interrupt signal is detected (KRM0n = 1) by selecting the falling edge (KRMD = 0), pull up the relevant input pins to V_{DD} by an external resistor. The internal pull-up resistor can be used by setting the relevant bits to 1 in the key interrupt input pins PU01 to PU04, PU40, and PU125 (pull-up resistor registers 0, 4, 12 (PU0, PU4, and PU12)).**

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.

3. The bits not used in the key interrupt mode can be used as normal ports.

12.3.3 Key return flag register (KRF)

This register controls the key return flags (KRF0 to KRF5).

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

Reset signal generation clears this register to 00H.

Figure 12-4. Format of Key Return Flag Register (KRF)

Address: FFF35H After reset: 00H R/W[Note]

Symbol	7	6	5	4	3	2	1	0
KRF	0	0	KRF5	KRF4	KRF3	KRF2	KRF1	KRF0

KRFn	Key interrupt flag
0	No key interrupt signal has been detected.
1	A key interrupt signal has been detected.

Note Writing to 1 is invalid. To clear KRFn, write “0” to the target bits and write “1” to other bits, with the 8-bit memory manipulation instruction.

<R> **Caution When the key interrupt flag is not used (KRMD = 0), accessing the KRF register is prohibited.**

12.3.4 Port mode registers 0, 4 (PM0, PM4)

These registers set the input and output of ports 0 and 4 in 1-bit units.

Set 1 to the bits corresponding with port registers 0, 4, and 12 (P0, P4, P12) corresponding to each port when using P01/KR2 to P04/KR5 and P40/KR0 as a key input.

In addition, Set 1 to the bit of port mode registers (PM0 and PM4).

The PM0 and PM4, registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to FFH.

Figure 12-5. Format of Port Mode Registers 0, 4 (PM0, PM4)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	1	1	PM04	PM03	PM02	PM01	1

PM0n	I/O mode selection for P0n/KRm pin (n = 1 to 4, m = 2 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	1	1	1	1	1	1	1	PM40

PM4n	I/O mode selection for P40/KR0 pin
0	Output mode (output buffer on)
1	Input mode (output buffer off)

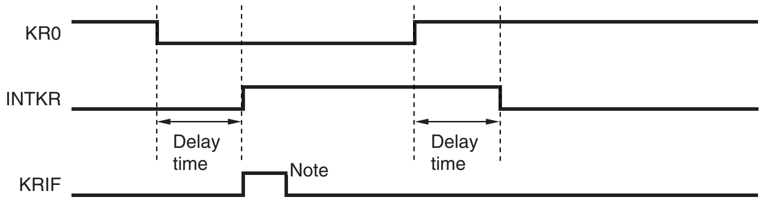
<R> 12.4 Key Interrupt Operation

12.4.1 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 KR5). 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 KR5).

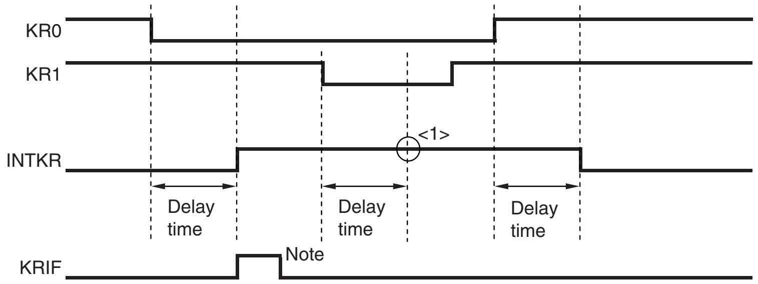
Figure 12-6. Operation of INTKR Signal When a Key Interrupt is Input to a Single Channel (When KRMD = 0 and KREG = 0)



Note 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 12-7 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 12-7. Operation of INTKR Signal When Key Interrupts Are Input to Multiple Channels (When KRMD = 0 and KREG = 0)



Note Acknowledgment of vectored interrupt request or bit cleared by software

12.4.2 When using the key interrupt flag (KRMD = 1)

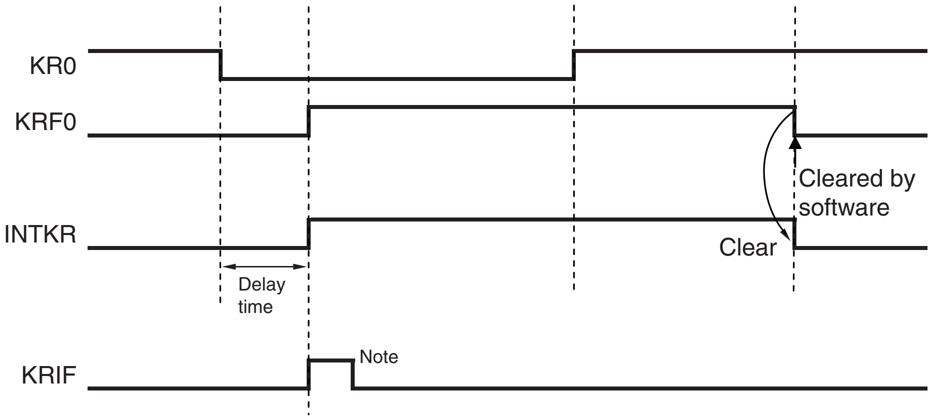
A key interrupt (INTKR) is generated when the valid edge specified by the setting of the KREG bit is input to a key interrupt pin (KR0 to KR5). The channels to which the valid edge was input can be identified by reading the key return flag register (KRF) after the key interrupt (INTKR) is generated.

If the KRMD bit is set to 1, the INTKR signal is cleared by clearing the corresponding bit in the KRF register.

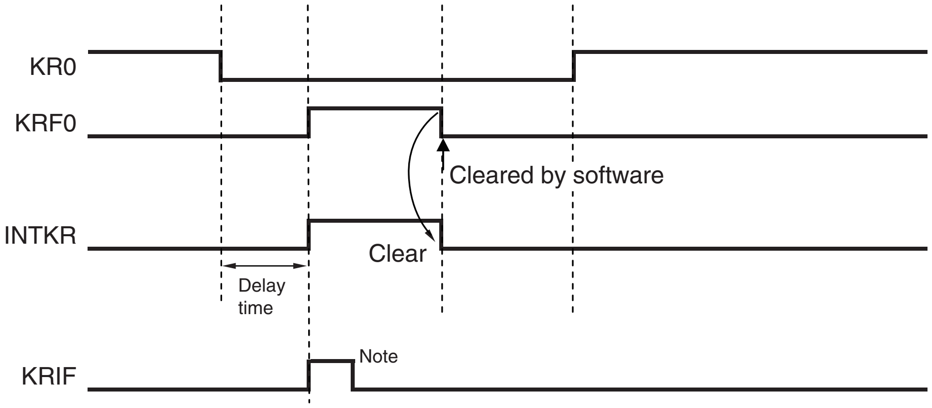
As shown in Figure 12-8, only one interrupt is generated each time a falling edge is input to one channel (when KREG = 0), regardless of whether the KRFn bit is cleared before or after a rising edge is input.

Figure 12-8. 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



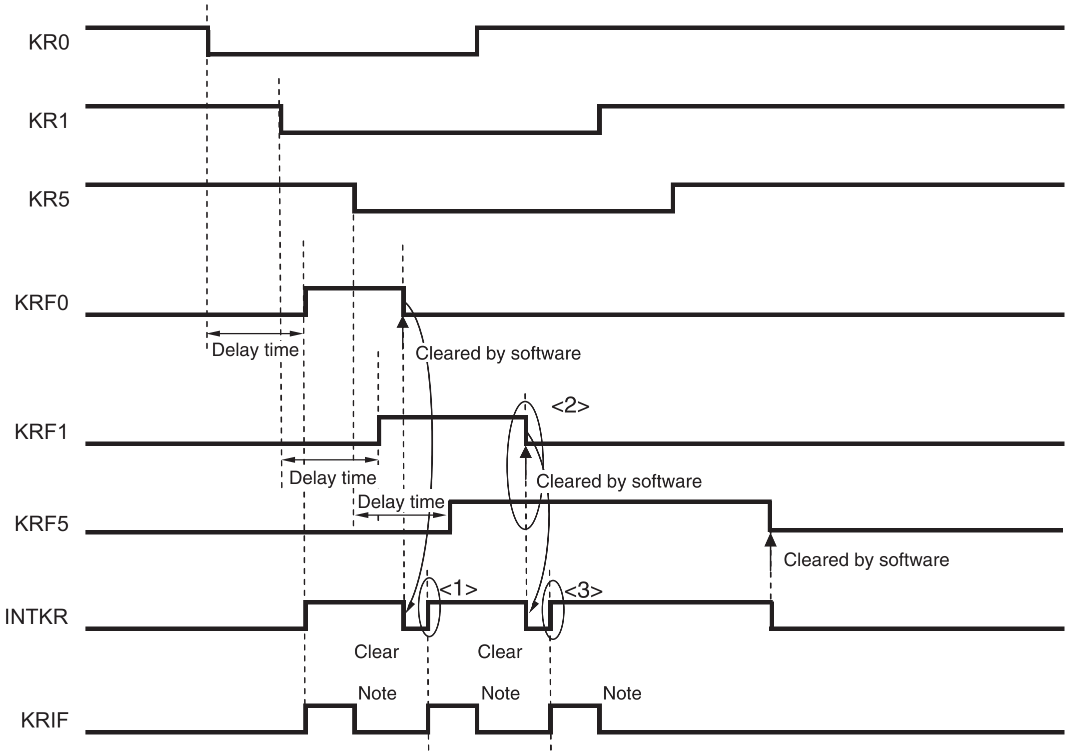
(b) When KRF0 is cleared before a rising edge is input to the KR0 pin



Note 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 12-9 below. A falling edge is also input to the KR1 and KR5 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 KR5 pin, the KRF5 bit is set (<2> in the figure) when the KRF1 bit is cleared. A key interrupt (INTKR) is therefore generated one clock (f_{CLK}) after the KRF1 bit is cleared (<3> in the figure). It is thus possible to generate a key interrupt (INTKR) when a valid edge is input to multiple channels.

Figure 12-9. Operation of INTKR Signal When Key Interrupts Are Input to Multiple Channels (When KRMD = 1 and KREG = 0)

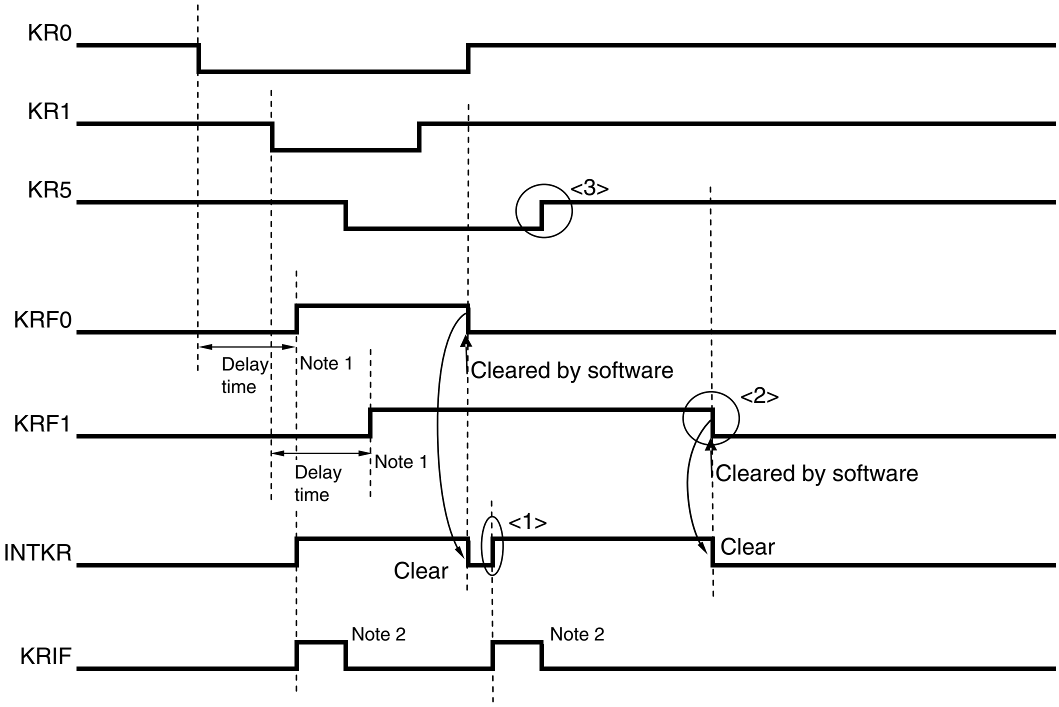


Note Acknowledgment of vectored interrupt request or bit cleared by software

Remark f_{CLK}: CPU/peripheral hardware clock frequency

The operation when a valid edge is input to the KR5 pins without generating a key interrupt (INTKR) is shown in Figure 12-10 below. A falling edge is also input to the KR1 and KR5 pins after a falling edge was input to the KR0 pin (when KREG = 0). The KR1 pin becomes high level when the KRF0 bit is cleared, but because the KRF1 bit is set, a key interrupt (INTKR) is generated one clock (f_{CLK}) after the KRF0 bit is cleared (<1> in the figure). Also, because the KR5 pin was high level (<3> in the figure) before the KRF1 bit was cleared (<2> in the figure) a key interrupt (INTKR) is not generated for the KR5 pin.

Figure 12-10. Operation When an INTKR Signal Is Not Generated upon Input of a Valid Edge to KR5 (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 **21.4 AC Characteristics**).

2. Acknowledgment of vectored interrupt request or bit cleared by software

Remark f_{CLK}: CPU/peripheral hardware clock frequency

CHAPTER 13 STANDBY FUNCTION

13.1 Overview

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 on-chip oscillator is operating before the HALT mode is set, oscillation of 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 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.

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 following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

2. 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 16 OPTION BYTE.

13.2 Standby Function Operation

13.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction.

The operating statuses in the HALT mode are shown below.

<R> **Caution Because the interrupt request signal is used to clear the HALT mode, if there is an interrupt source with the interrupt request flag set (1) and the interrupt mask flag cleared (0), the HALT mode is not entered.**

Table 13-1. Operating Statuses in HALT Mode

HALT Mode Setting Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock
			When CPU Is Operating on High-speed On-chip Oscillator Clock (f_{IH})
System clock			Clock supply to the CPU is stopped
	High-speed on-chip oscillator clock	f_{IH}	Operation continues (cannot be stopped)
	Low-speed on-chip oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1, and WDSTBYON = 1: Oscillates • WDTON = 1, and WDSTBYON = 0: Stops
CPU			Operation stopped
Code flash memory			Operation stopped
RAM			
Port (latch)			Status before HALT mode was set is retained
Timer array unit			Operable
Watchdog timer			Set by bit 0 (WDSTBYON) of option byte (000C0H) • WDSTBYON = 0: Operation stopped • WDSTBYON = 1: Operation continues
Clock output/buzzer output			Operable
A/D converter			
Serial array unit (SAU)			
Selectable power-on-reset function			
External interrupt			
Key interrupt 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.

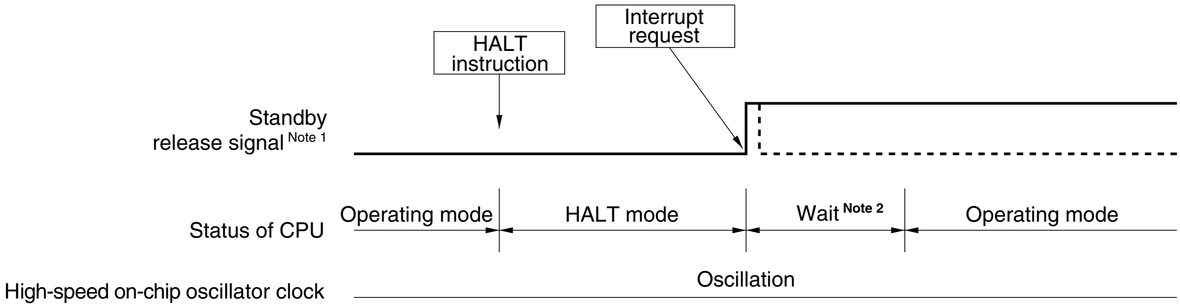
(2) HALT mode release

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 13-1. HALT Mode Release by Interrupt Request Generation



<R> **Notes 1.** For details of the standby release signal, Refer to **Figure 11-1 Basic Configuration of Interrupt Function**.

<R> **2.** Wait time inserted until HALT mode release

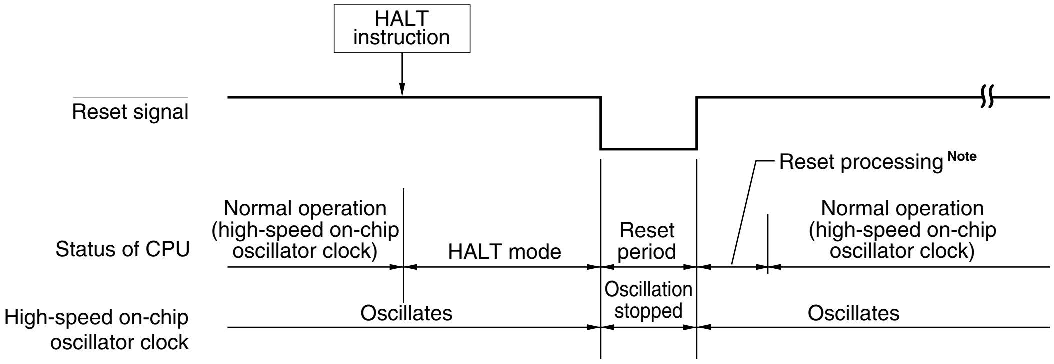
- When vectored interrupt servicing is carried out: 28 to 29 clocks
- When vectored interrupt servicing is not carried out: 20 to 21 clcoks

Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) HALT mode 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 13-2. HALT Mode Release by Reset Signal Generation



<R> **Note** For the reset processing time, see **CHAPTER 14 RESET FUNCTION**. For the reset processing time of the SPOR circuit, see **CHAPTER 15 SELECTABLE POWER-ON-RESET CIRCUIT**.

13.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction.

Caution If there is an interrupt source whose corresponding interrupt request flag is set (1) and interrupt mask flag is cleared (0), the interrupt request signal is used to release the STOP mode. If the STOP mode is set, therefore, it is only exited after oscillation stabilizes.

The operating statuses in the STOP mode are shown below.

Table 13-2. Operating Statuses in STOP Mode

STOP Mode Setting Item			When STOP Instruction Is Executed While CPU Is Operating
			When CPU Is Operating on High-speed On-chip Oscillator Clock (f_{IH})
System clock			Clock supply to the CPU is stopped
	High-speed on-chip oscillator clock	f_{IH}	Stopped
	Low-speed on-chip oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1, and WDSTBYON = 1: Oscillates • WDTON = 1, and WDSTBYON = 0: Stops
CPU			Operation stopped
Code flash memory			
RAM			Operation stopped
Port (latch)			Status before STOP mode was set is retained
Timer array unit			Operation disabled
Watchdog timer			Set by bit 0 (WDSTBYON) of option byte (000C0H) • WDSTBYON = 0: Operation stopped • WDSTBYON = 1: Operation continues
Clock output/buzzer output			Operation prohibited
A/D converter			
Serial array unit (SAU)			
Selectable power-on-reset function			Operable
External interrupt			
Key interrupt 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.

Cautions 1. To use the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.

2. To stop the low-speed on-chip oscillator clock in the STOP mode, must previously be set an option byte to stop the watchdog timer operation in the HALT/STOP mode (bit 0 (WDSTBYON) of 000C0H = 0).

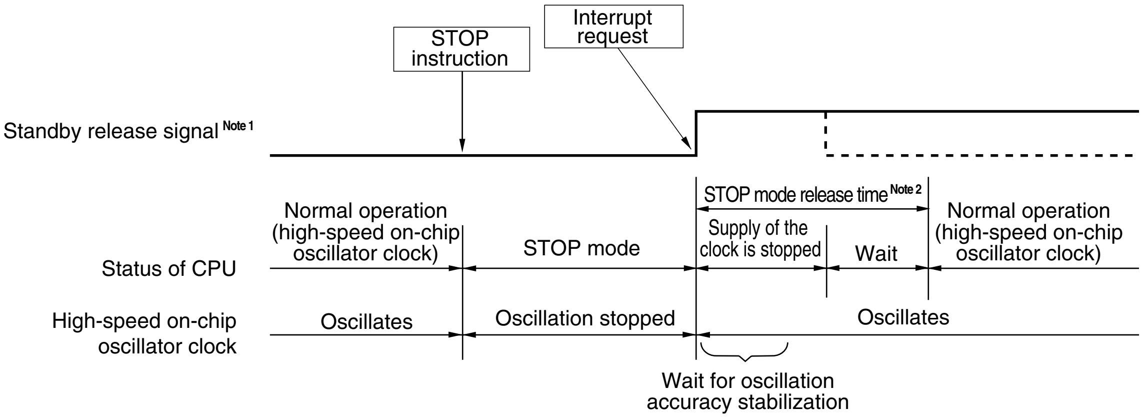
(2) STOP mode release

The STOP mode can be released by the following two sources.

(a) STOP mode 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 13-3. STOP Mode Release by Interrupt Request Generation



<R> **Notes 1.** For details of the standby release signal, see **Figure 11-1. Basic Configuration of Interrupt Function**.

<R> **2.** STOP mode release time

Supply of the clock is stopped: 27 μs (TYP.)

Wait time inserted until STOP mode release

- When vectored interrupt servicing is carried out: 11 clocks
- When vectored interrupt servicing is not carried out: 3 clocks

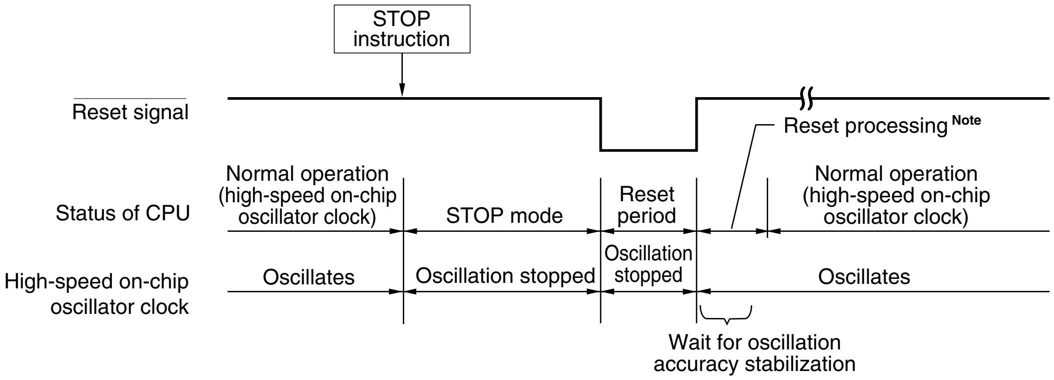
<R> **Remarks 1.** The clock supply stop time varies depending on the temperature conditions and STOP mode period.

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 13-4. STOP Mode Release by Interrupt Request Generation



Note For the reset processing time, see **CHAPTER 14 RESET FUNCTION**.
For the reset processing time of the SPOR circuit, see **CHAPTER 15 SELECTABLE POWER-ON-RESET CIRCUIT**.

CHAPTER 14 RESET FUNCTION

The following four 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 selectable power-on-reset (SPOR) circuit
(4) Internal reset by execution of illegal instruction$^{\text{Note 1}}$
(5) Internal reset by the data retention voltage$^{\text{Note 2}}$

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.

Notes1. This reset occurs when instruction code FFH is executed.
This reset does not occur during emulation using an in-circuit emulator or an on-chip debugging emulator.
2. Data is not reset while V_{DD} is greater than or equal to the data retention voltage. Data is reset when V_{DD} falls below the data retention voltage. The maximum voltage at which data is reset is prescribed as the data retention voltage specification.

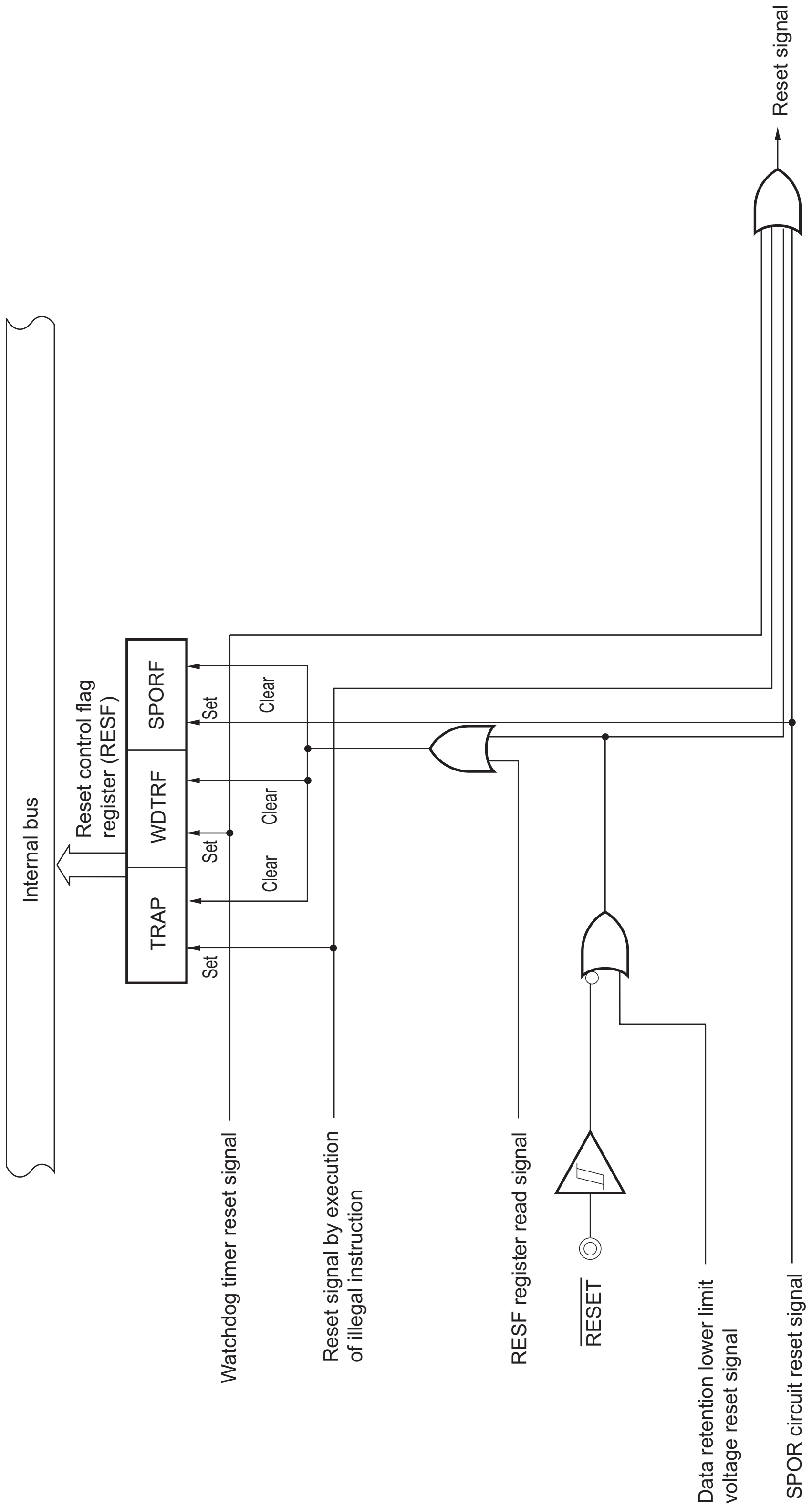
<R> **Cautions 1. For an external reset, set the PORTSELB bit of the user option byte (000C1H) to 1 so that the P125 pin operates as $\overline{\text{RESET}}$, and input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.**
(To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, and then apply power supply. The $\overline{\text{RESET}}$ pin must be kept low for at least 10 μs during the period in which the supply voltage is within the operating range shown in 21.3 AC Characteristics before inputting a high level to the $\overline{\text{RESET}}$ pin.)

<R> **2. During reset input, the high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating.**

<R> **3. The port pin becomes the following status because each SFR and 2nd SFR are initialized after reset.**

- **P40: High-impedance during external reset period or reset period by the data retention power supply voltage. High level during other types of reset or after receiving a reset (connected to the internal pull-up resistor).**
- **P125: Low level during external reset period (low level input to $\overline{\text{RESET}}$ pin). High level during other types of reset period or after receiving a reset (connected to the internal pull-up resistor).**
- **Ports other than P40 and P125: High-impedance during reset period or after receiving a reset.**

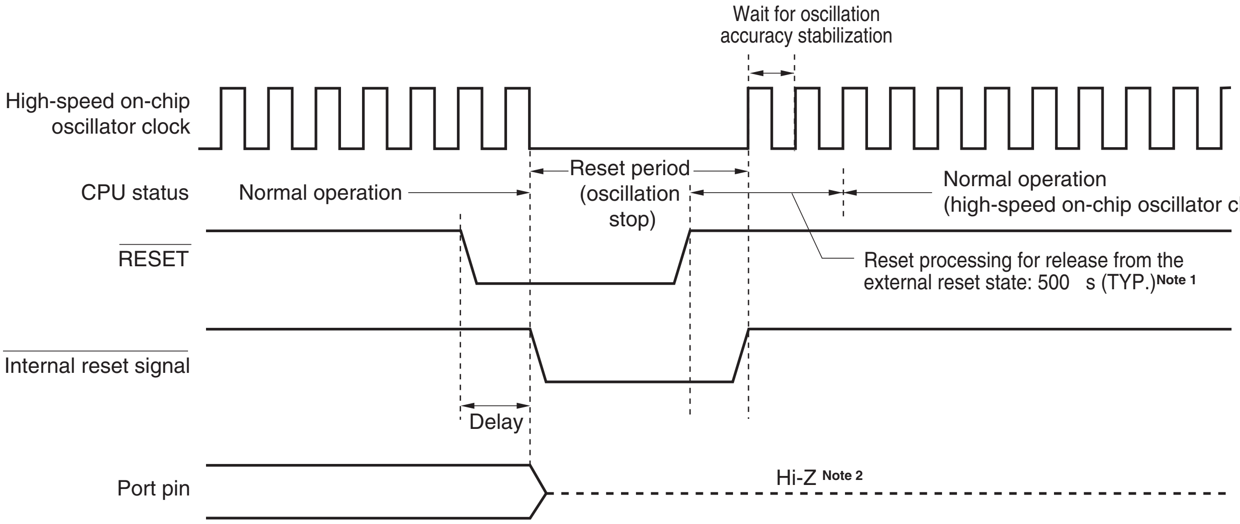
Figure 14-1. Block Diagram of Reset Function



14.1 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 14-2. Timing of Reset by $\overline{\text{RESET}}$ Input



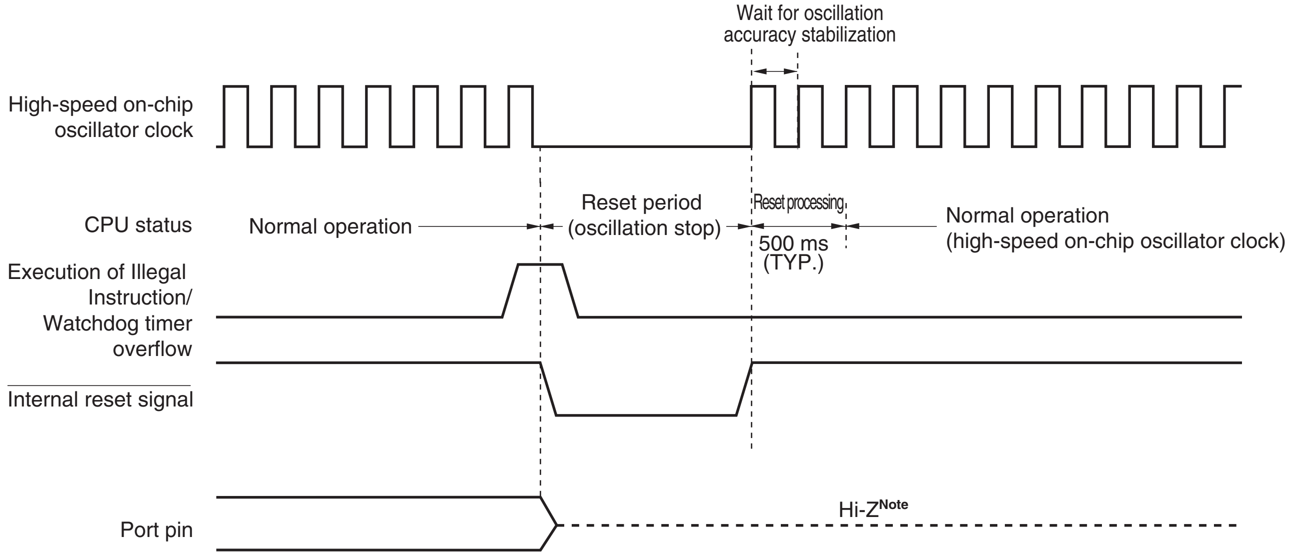
<R> **Notes 1.** After power is supplied, an SPOR reset processing time of (MAX. 3.39 ms) is required before reset processing starts after release of the external reset.

<R> **2.** Status of port pin P40 is as follows.

- High-impedance during external reset period or reset period by the data retention power supply voltage
- High level after receiving a reset (connected to the internal pull-up resistor)

Release from the reset state is automatic in the cases of a reset due to the watchdog timer overflow or a reset due to the execution of an illegal instruction. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 14-3. Timing of Reset Due to Watchdog Timer Overflow or Execution of Illegal Instruction



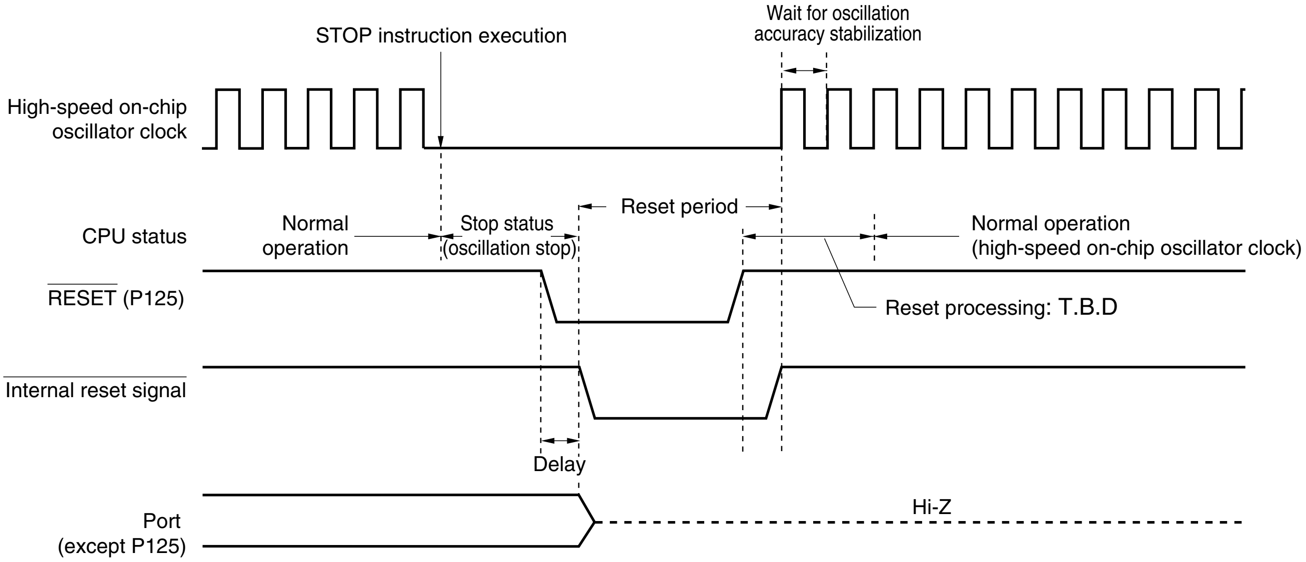
<R> **Note** Statuses of port pins P40 and P125 pins are as follows.

- High level during reset period or after receiving a reset (connected to the internal pull-up resistor).

Caution A watchdog timer internal reset resets the watchdog timer.

Remark For the reset timing due to the voltage detection by the selectable power-on-reset (SPOR) circuit, see **CHAPTER 15 SELECTABLE POWER-ON-RESET CIRCUIT**.

Figure 14-4. Timing of Reset in STOP Mode by RESET Input



14.2 Operation States During Reset Periods

Table 14-1 shows the operation states during reset periods. Table 14-2 shows the state of the hardware after acceptance of a reset.

Table 14-1. Operation States During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	High-speed on-chip oscillator clock	f_{IH}	Operation stopped
	Low-speed on-chip oscillator clock	f_{IL}	
CPU			Operation stopped
Code flash memory			Operation stopped
RAM			Operation stopped
Port (latch)			High impedance[Note]
Timer array unit			Operation stopped
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Serial array unit (SAU)			
Selectable power-on-reset function			Detection operation possible
External interrupt			Operation stopped
Key interrupt function			

Note Statuses of P40 and P125 pins are as follows

- P40: High-impedance during external reset period or reset period by the data retention power supply voltage. High level during other types of reset period or after receiving a reset (connected to the internal pull-up resistor).
- P125: Low level during external reset period (low level input to $\overline{\text{RESET}}$ pin). High level during other types of reset period or after receiving a reset (connected to the internal pull-up resistor).

Table 14-2. State of Hardware After Acceptance of Reset (1/3)

Hardware		After Acceptance of Reset[Note 1]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined
	General-purpose registers	Undefined
Port registers (P0, P4, P12, P13 (output latches))		P0: 00H, P4: 01H, P12, P13: Undefined
Port mode registers (PM0, PM4)		FFH
Port mode control register (PMC0)		FFH
Port output mode register (POM0)		00H
Pull-up resistor option registers (PU0, PU4, PU12)		PU0: 00H, PU4: 01H, PU12: 20H
Peripheral I/O redirection register (PIOR)		00H
Noise filter enable registers 0, 1 (NFEN0, NFEN1)		00H
Peripheral enable register 0 (PER0)		00H
High-speed on-chip oscillator frequency select register (HOCODIV)		Undefined
Timer array unit	Timer data register 00H/L (TDR00H, TDR00L)	00H
	Timer data register 01H/L (TDR01H, TDR01)	00H
	Timer mode register 00H/L(TMR00H, TMR00L)	00H
	Timer mode register 01H/L(TMR01H, TMR01L)	00H
	Timer status register 00 (TSR00)	00H
	Timer status register 01 (TSR01)	00H
	Timer counter register 00H/L (TCR00H, TCR00L)	FFH
	Timer counter register 01H/L (TCR01H, TCR01L)	FFH
	Timer channel enable status register 0 (TE0, TEH0)	00H
	Timer channel start register 0 (TS0, TSH0)	00H
	Timer channel stop register 0 (TT0, TTH0)	00H
	Timer clock select register 0 (TPS0)	00H
	Timer output register 0 (TO0)	00H
	Timer output enable register 0 (TOE0)	00H
	Timer output level register 0 (TOL0)	00H
	Timer output mode register 0 (TOM0)	00H
Clock output/buzzer output	Clock output select register 0 (CKS0)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 2]

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. The reset value of WDTE is decided by the setting of the option byte (WDTON bit).

Table 14-2. State of Hardware After Acceptance of Reset (2/3)

Hardware		After Acceptance of Reset[Note]
A/D converter	A/D conversion result lower bit storage register (ADCRL)	00H
	A/D conversion result higher bit storage register (ADCRH)	00H
	A/D converter mode registers 0, 2 (ADM0, ADM2)	00H
	Analog input channel specification register (ADS)	00H
Serial array unit (SAU)	Serial data registers 00H/L (SDR00H, SDR00L)	00H
	Serial data registers 01H/L (SDR01H, SDR01L)	00H
	Serial status registers 00, 01 (SSR00, SSR01)	00H
	Serial flag clear trigger registers 00, 01 (SIR00, SIR01)	00H
	Serial mode registers 00H, 01H (SMR00H, SMR01H)	00H
	Serial mode registers 00L, 01L (SMR00L, SMR01L)	20H
	Serial communication operation setting registers 00H, 01H (SCR00H, SCR01H)	00H
	Serial communication operation setting registers 00L, 01L (SCR00L, SCR01L)	87H
	Serial channel enable status register 0 (SE0)	00H
	Serial channel start register 0 (SS0)	00H
	Serial channel stop register 0 (ST0)	00H
	Serial clock select register 0 (SPS0)	00H
	Serial output register 0 (SO0)	03H
	Serial clock output register 0 (CKO0)	03H
	Serial output enable register 0 (SOE0)	00H
	Serial output level register 0 (SOL0)	00H
Key interrupt	Key return control register (KRTCL)	00H
	Key return mode register (KRM0)	00H
	Key return flag register (KRF)	00H

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Table 14-2. State of Hardware After Acceptance of Reset (3/3)

Hardware		After Acceptance of Reset[Note 1]
Reset function	Reset control flag register (RESF)	**Note 2**
Interrupt	Request flag registers 0L, 0H (IF0L, IF0H)	00H
	Mask flag registers 0L, 0H (MK0L, MK0H)	FFH
	Priority specification flag registers 00L, 00H, PR10L, PR10H, (PR00L, PR00H, PR10L, PR10H)	FFH
	External interrupt rising edge enable register 0 (EGP0)	00H
	External interrupt falling edge enable register 0 (EGN0)	00H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. These values vary depending on the reset source.

Reset Source Register		$\overline{\text{RESET}}$ Input	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by SPOR	Reset by data retention power supply voltage[Note]
RESF	TRAP bit	Cleared (0)	Set (1)	Held	Held	Cleared (0)
	WDTRF bit		Held	Set (1)		
	SPORF bit		Held	Held	Set (1)	

Note Data is not reset while V_{DD} is greater than or equal to the data retention voltage. Data is reset when V_{DD} falls below the data retention voltage. The maximum voltage at which data is reset is prescribed as the data retention voltage specification.

14.3 Register for Confirming Reset Source

14.3.1 Reset control flag register (RESF)

Many internal reset generation sources exist in the RL78 Microcontroller. The reset control flag register (RESF) is used to store which source has generated the reset request.

The RESF register can be read by an 8-bit memory manipulation instruction.

The external reset, a reset by the data retention lower limit voltage, and reading the RESF register clear TRAP, WDTRF, and SPORF flags.

Figure 14-5. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: Note 1 R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	0	0	SPORF

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.

SPORF	Internal reset request by selectable power-on reset (SPOR) circuit
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 source.

2. This reset occurs when instruction code FFH is executed.
This reset does not occur during emulation using an in-circuit emulator or an on-chip debugging emulator.

Caution Do not read data by a 1-bit memory manipulation instruction.

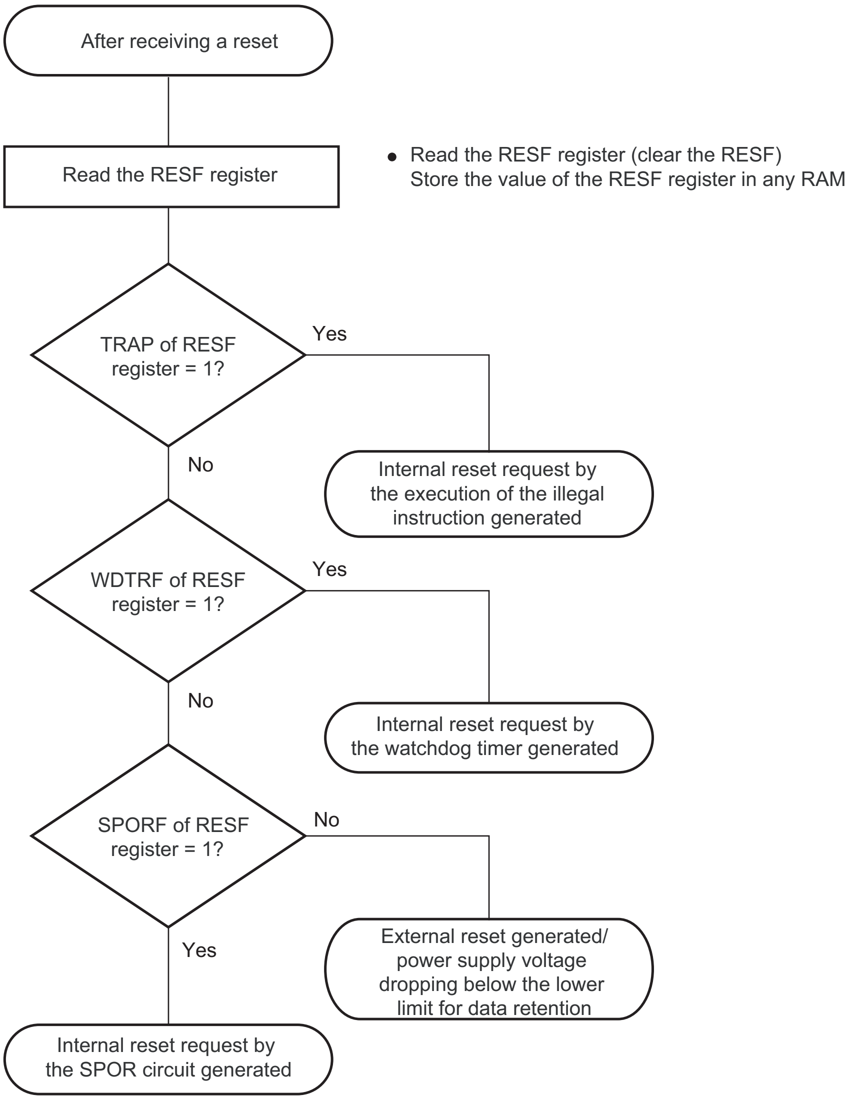
The status of the RESF register when a reset request is generated is shown in Table 14-3.

Table 14-3. RESF Register Status When Reset Request Is Generated

Reset Source / Flag	$\overline{\text{RESET}}$ Input	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by SPOR	Reset by data retention lower limit voltage
TRAP bit	Cleared (0)	Set (1)	Held	Held	Cleared (0)
WDTRF bit		Held	Set (1)	Held	
SPORF bit		Held	Held	Set (1)	

The RESF register is automatically cleared when it is read by an 8-bit memory manipulation instruction. Figure 14-6 shows the procedure for checking the reset source.

Figure 14-6. Procedure for Checking Reset Source



CHAPTER 15 SELECTABLE POWER-ON-RESET CIRCUIT

<R> 15.1 Functions of Selectable Power-on-reset Circuit

The selectable power-on-reset (SPOR) circuit has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage exceeds the detection voltage ($V_{DD} \geq V_{SPOR}$).
- The SPOR circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{SPDR}), and generates an internal reset signal when $V_{DD} < V_{SPDR}$.
- The detection level for the power supply detection voltage (V_{SPOR}, V_{SPDR}) can be selected by using the option byte (000C1H) as one of 4 levels (for details, see **16.2 Format of User Option Byte**).

Bit 0 (SPORF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 14 RESET FUNCTION**.

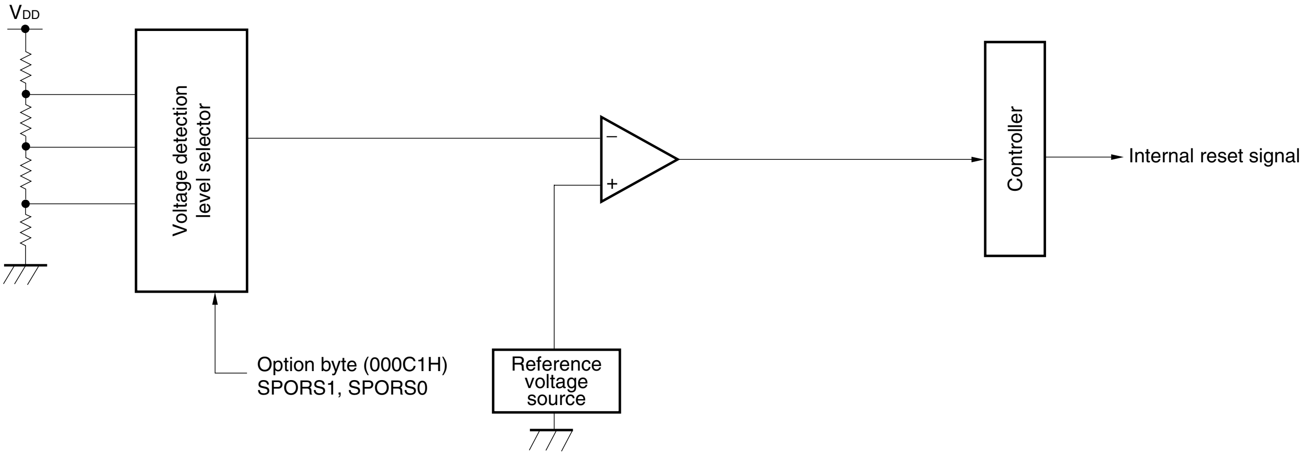
Caution The values of all flags in the reset control flag register (RESF) are retained until V_{DD} reaches data retention lower limit voltage.

Remark V_{SPOR}: SPOR power supply rise detection voltage
V_{SPDR}: SPOR power supply fall detection voltage
For details, see **21.6.3 SPOR circuit characteristics**.

15.2 Configuration of Selectable Power-on-reset Circuit

The block diagram of the selectable power-on-reset circuit is shown in Figure 15-1.

<R> **Figure 15-1. Block Diagram of Selectable Power-on-reset Circuit**



<R> 15.3 Operation of Selectable Power-on-reset Circuit

Specify the voltage detection level by using the option byte 000C1H.

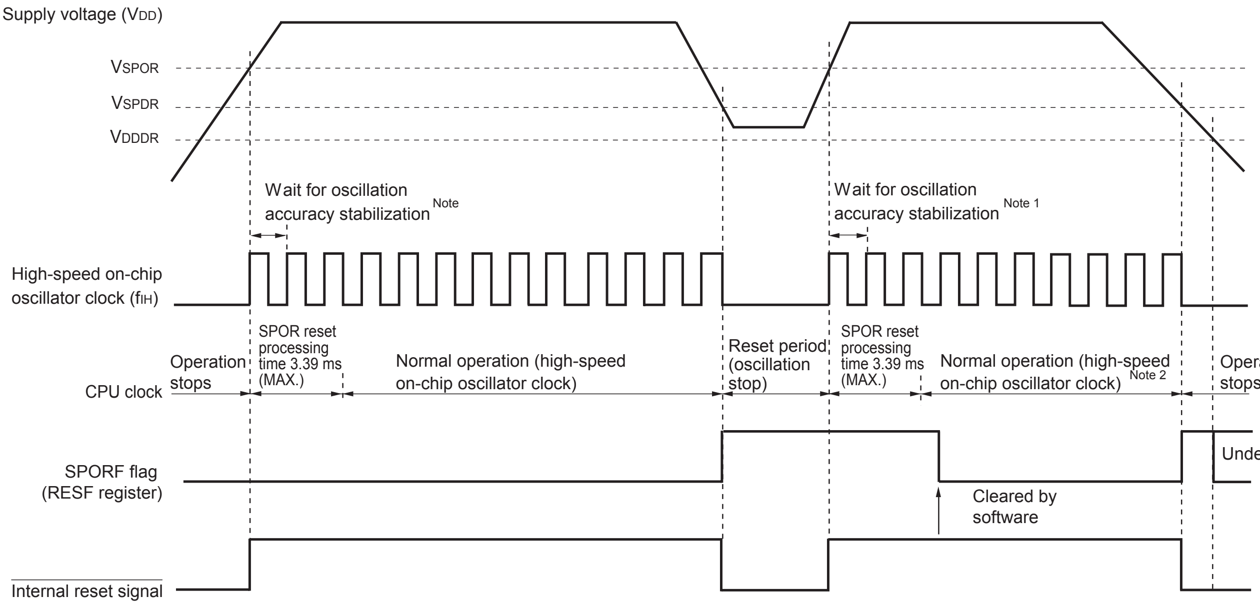
The internal reset signal is generated at power on.

The internal reset status is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{SPOR}). The internal reset is cleared when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{SPOR}).

The internal reset is generated when the supply voltage (V_{DD}) drops lower than the voltage detection level (V_{SPDR}).

Figure 15-2 shows the timing of the internal reset signal generated by the selectable power-on-reset circuit.

Figure 15-2. Timing of Internal Reset Signal Generation



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

Remark
V_{SPOR}: SPOR power supply rise detection voltage
V_{SPDR}: SPOR power supply fall detection voltage
V_{DDDR}: Data retain power supply voltage

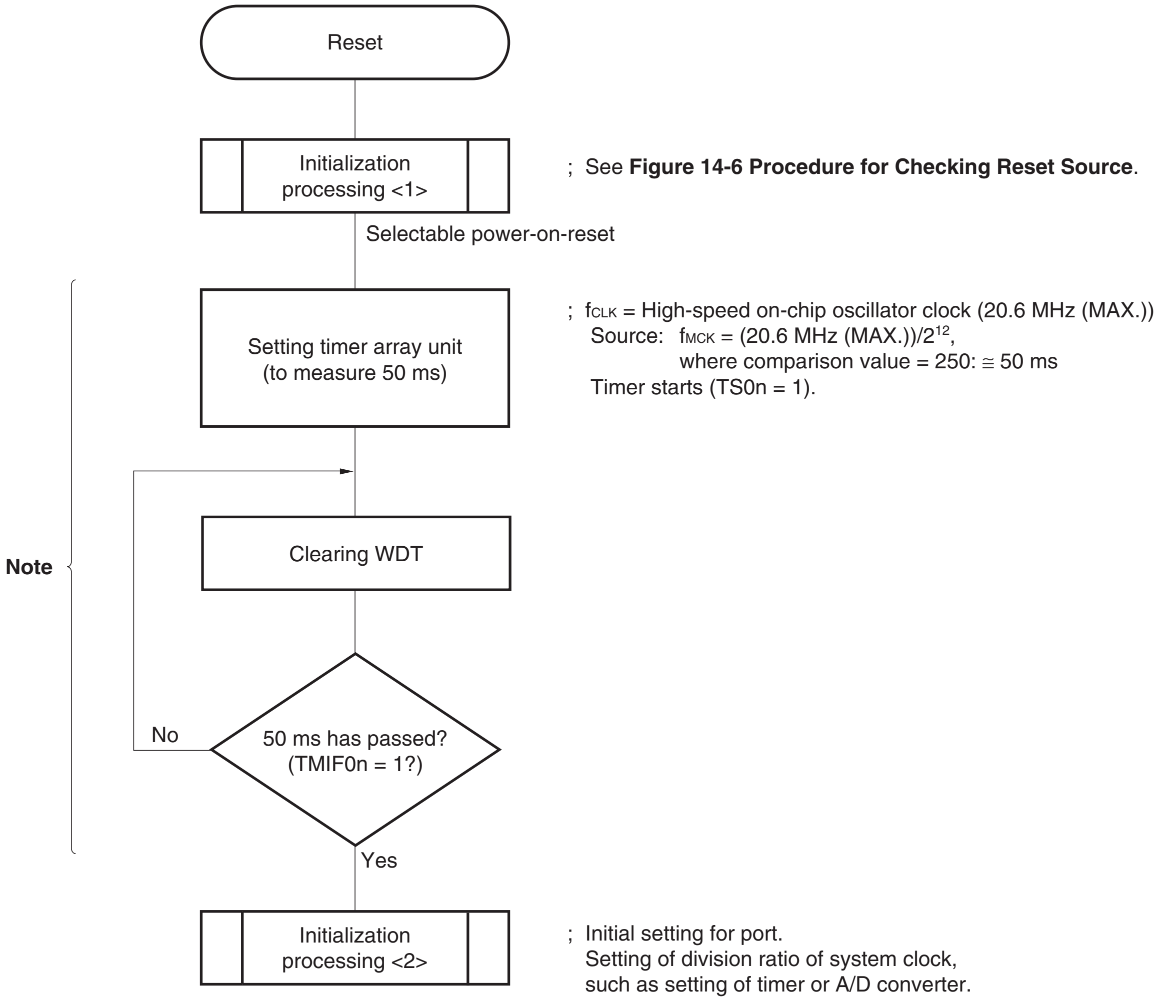
<R> 15.4 Cautions for Selectable Power-on-reset Circuit

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the SPOR detection voltage (V_{SPOR}, V_{SPDR}), 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 timer and etc., and then initialize the ports.

Figure 15-3. Example of Software Processing When Supply Voltage Fluctuation is 50 ms or Less in Vicinity of the Voltage Detection Level



Note If reset is generated again during this period, initialization processing <2> is not started.

Remark n: Channel number (n = 0, 1)

CHAPTER 16 OPTION BYTE

16.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the R7F0C80112ESP, R7F0C80212ESP 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.

The bits to which no function is allocated must be used with their initial value.

16.1.1 User option byte (000C0H to 000C2H)

(1) 000C0H

○ Operation of watchdog timer
- Counter operation is enabled or disabled.
- Operation is stopped or enabled in the HALT or STOP mode.

○ Time setting of watchdog timer
- Setting of overflow time of watchdog timer
- Setting of interval interrupt time of watchdog timer

(2) 000C1H

○ Setting of SPOR detection level (V_{SPOR})

○ Controlling of P125/KR1/$\overline{\text{RESET}}$ pin
- P125/KR1 or $\overline{\text{RESET}}$

(3) 000C2H

○ Setting of the frequency of the high-speed on-chip oscillator
- Select from 1.25 to 20 MHz.

16.1.2 On-chip debug option byte (000C3H)

○ Control of on-chip debug operation
- On-chip debug operation is disabled or enabled.

16.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 16-1. Format of User Option Byte (000C0H)

Address: 000C0H

7	6	5	4	3	2	1	0
1	1	1	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

<R>

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (WDTRES)	Watchdog timer interval interrupt time (Count value of overflow x3/4+3/(f_{IL}x4))
0	0	0	$(2^6 -1)/f_{IL}$ (2.1 ms)	1.6 ms
0	0	1	$(2^7 -1)/f_{IL}$ (4.23 ms)	3.2 ms
0	1	0	$(2^8 -1)/f_{IL}$ (8.5 ms)	6.4 ms
0	1	1	$(2^9 -1)/f_{IL}$ (17.03 ms)	12.8 ms
1	0	0	$(2^{11} -1)/f_{IL}$ (68.23 ms)	51.2 ms
1	0	1	$(2^{13} -1)/f_{IL}$ (273.03 ms)	204.8 ms
1	1	0	$(2^{14} -1)/f_{IL}$ (546.1 ms)	409.6 ms
1	1	1	$(2^{16} -1)/f_{IL}$ (2184.5 ms)	1638.4 ms

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode
1	Counter operation enabled in HALT/STOP mode

Cautions 1. Be sure to write 1 to bits 7 to 5.

2. Setting WDTON = 0 and WDSTBON = 1 is prohibited.

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

Figure 16-2. Format of User Option Byte (000C1H)

Address: 000C1H

7	6	5	4	3	2	1	0
1	1	1	PORTSELB	SPORS1	SPORS0	1	1

- Setting of SPOR detection voltage

Detection voltage		Option byte setting value	
V_{SPOR}	V_{SPDR}	SPORS1	SPORS0
Rising edge	Falling edge		
4.28 V	4.00 V	0	0
2.90 V	2.70 V	0	1
2.57 V	2.40 V	1	0
Other than above		Setting prohibited	

- P125/$\overline{\text{RESET}}$ pin control

PORTSELB	P125/$\overline{\text{RESET}}$ pin control
0	Port function (P125/KR1)
1	$\overline{\text{RESET}}$ input (internal pull-up resistor can be always connected.)

Caution Be sure to write 1 to bits 7 to 5, 1, and 0.

<R> **Remarks 1.** For details on the SPOR circuit, see **CHAPTER 15 SELECTABLE POWER-ON-RESET CIRCUIT**.

2. The detection voltage for the rising edge indicates the typical value, and the detection voltage for the falling edge indicates the minimum value.. For details, see **21.6.3 SPOR circuit characteristics**.

Figure 16-3. Format of User Option Byte (000C2H)

Address: 000C2H

7	6	5	4	3	2	1	0
1	1	1	1	1	FRQSEL2	FRQSEL1	FRQSEL0

FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
0	0	1	20 MHz
0	1	0	10 MHz
0	1	1	5 MHz
1	0	0	2.5 MHz
1	0	1	1.25 MHz
Other than above			Setting prohibited

Caution Be sure to write 1 to bits 7 to 3.

16.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 16-4. 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	1

OCDENSET	Control of on-chip debug operation
0	Disables on-chip debug operation.
1	Enables on-chip debugging.Note

Note Does not erase data of flash memory in case of failures in authenticating on-chip debug security ID.

Caution Bit 7 (OCDENSET) can only be specified a value.
Be sure to set 0000101B to bits 6 to 0.

Remark The value on bits 3 and 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.

16.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the link option in addition to describing to the source.

When doing so, the contents set by using the link option take precedence, even if descriptions exist in the source, as mentioned below.

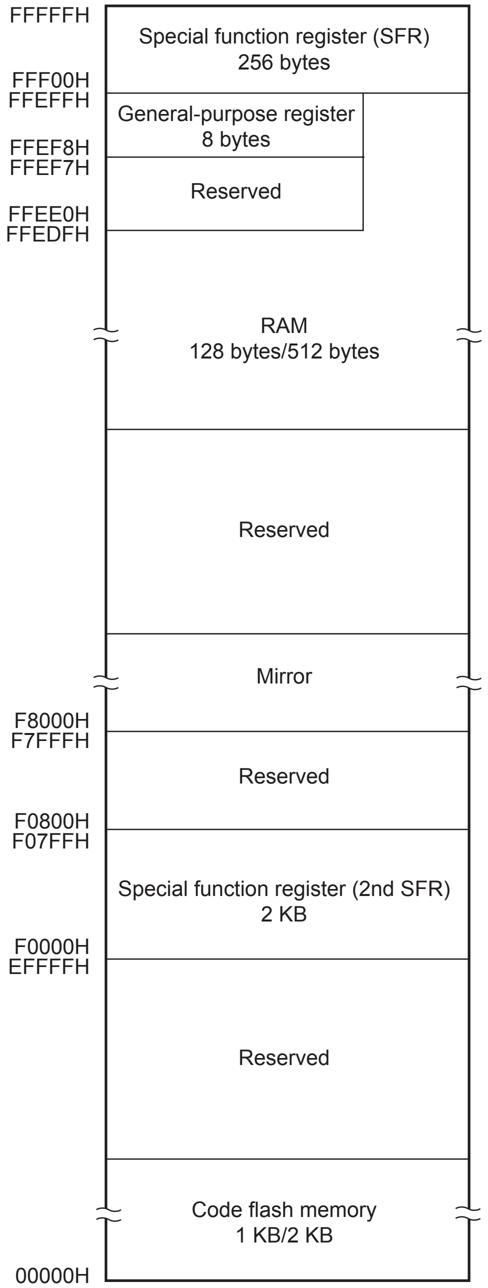
A software description example of the option byte setting is shown below.

```
    OPT   CSEG          OPT_BYTE
          DB            F7H       ;  Enables watchdog timer operation,
                                  ;  Overflow time of watchdog timer is 2^9/fIL,
                                  ;  Stops watchdog timer operation during HALT/STOP mode
          DB            E7H       ;  Select 2.70 V for VSPDR and 2.90 V for VSPOR
                                  ;  Use the port function (P125/KR1)
          DB            FDH       ;  Select 1.25 MHz as the frequency of the high-speed on-chip oscillator clock
          DB            85H       ;  Enables on-chip debug operation
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction.

CHAPTER 17 FLASH MEMORY

The RL78 microcontroller incorporates the flash memory to which a program can be written, erased, and overwritten.



The methods for programming the flash memory are as follows.

The contents of the code flash memory can be rewritten by serial programming using a flash memory programmer or an external device (UART communication).

- Serial programming by using a flash memory programmer (see **17.1**)
  Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.
- Serial programming by using an external device (UART communication) (see **17.2**)
  Data can be written to the flash memory on-board, by using UART communication with an external device (a microcontroller or ASIC).

17.1 Serial Programming by Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the RL78 microcontroller.

- PG-FP5, FL-PR5
- E1 on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the RL78 microcontroller has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the RL78 microcontroller is mounted on the target system.

Remark FL-PR5 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

Table 17-1. Wiring Between R7F0C80112ESP, R7F0C80212ESP and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.
Signal Name		I/O	Pin Function		
PG-FP5, FL-PR5	E1 On-chip Debugging Emulator				
–	TOOL0	I/O	Transmit/receive signal	TOOL0/P40	1
SI/RxD	–	I/O	Transmit/receive signal		
–	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	2
/RESET	–	Output			
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	5
GND		–	Ground	V_{SS}	4
EMV_{DD}		–	Driving power for TOOL pin	V_{DD}	5

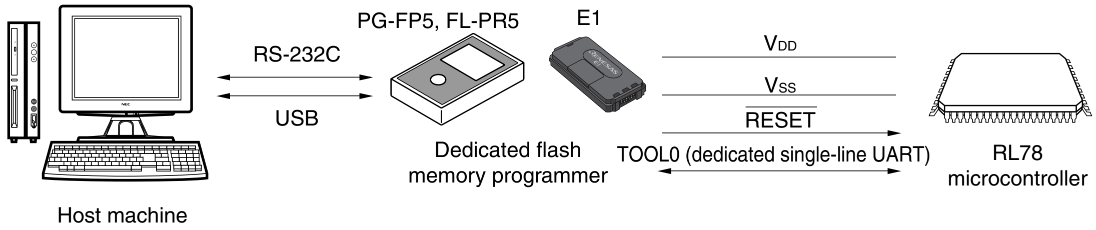
Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

About a connection RL78 microcontroller and a connector, refer to the user's manual of each programmer. About a connection with E1, refer to **18.1 Connecting E1 On-chip Debugging Emulator**..

17.1.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 17-1. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

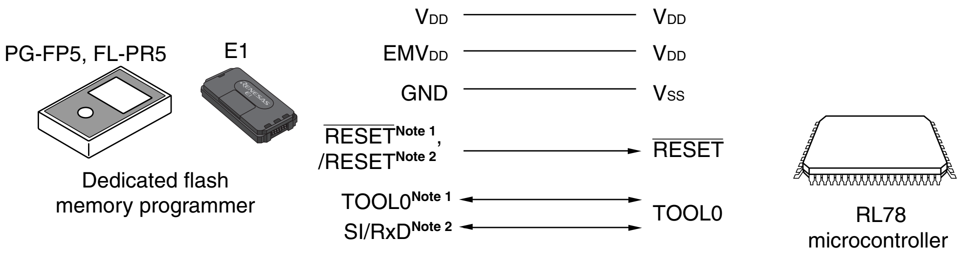
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

17.1.2 Communication mode

Communication between the dedicated flash memory programmer and the RL78 microcontroller is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78 microcontroller.

Transfer rate: Fixed to 115.2 kbps

Figure 17-2. Communication with Dedicated Flash Memory Programmer



Notes 1. When using E1 on-chip debugging emulator.

2. When using PG-FP5 or FL-PR5.

The dedicated flash memory programmer generates the following signals for the RL78 microcontroller. See the manuals of PG-FP5, FL-PR5, or E1 on-chip debugging emulator for details.

Table 17-2. Pin Connection

Dedicated Flash Memory Programmer				RL78 Microcontroller
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}
EMV_{DD}		–	Driving power for TOOL0 pin	V_{DD}
/RESET	–	Output	Reset signal	$\overline{\text{RESET}}$
–	$\overline{\text{RESET}}$	Output		
–	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	–	I/O	Transmit/receive signal	

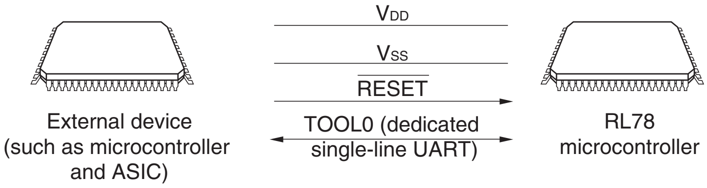
<R> 17.2 Writing to Flash Memory by Using External Device (that Incorporates UART)

On-board data writing to the internal flash memory is possible by using the RL78 microcontroller and an external device (a microcontroller or ASIC) connected to a UART.

17.2.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 17-3. Environment for Writing Program to Flash Memory



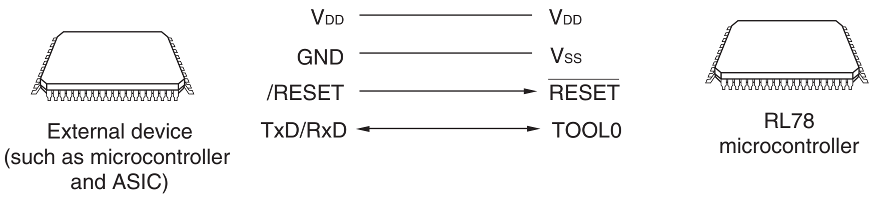
Processing to write data to or delete data from the RL78 microcontroller by using an external device is performed on-board. Off-board writing is not possible.

17.2.2 Communication Mode

Communication between the external device and the RL78 microcontroller is established by serial communication using the TOOL0 pin via the dedicated UART of the RL78 microcontroller.

Transfer rate: Fixed to 115.2 kbps

Figure 17-4. Communication with External Device



The external device generates the following signals for the RL78 microcontroller.

Table 17-4. Pin Connection

External Device			RL78 microcontroller
Signal Name	I/O	Pin Function	Pin Name
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND	–	Ground	V_{SS}
RESETOUT	Output	Reset signal output	$\overline{\text{RESET}}$
RxD	Input	Receive signal	TOOL0
TxD	Output	Transmit signal	

17.3 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

Remark Refer to flash programming mode, see **17.4.2 Flash memory programming mode**.

17.3.1 P40/TOOL0 pin

In the flash memory programming mode, pull up externally with a 1 kΩ resister, and connect it to the dedicated flash memory programmer.

When the P40/TOOL0 pin is in use as a port pin, if release from the reset state proceeds while the level on the P40 pin is low, the reset processing time increases by several hundred ms and the value returned in RESF on release from the reset state is 10H.

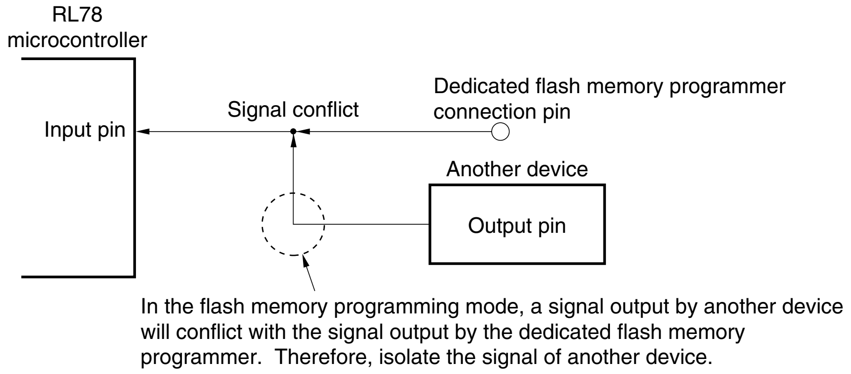
Remark The SAU pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

17.3.2 $\overline{\text{RESET}}$ 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 17-5. Signal Conflict ($\overline{\text{RESET}}$ Pin)



17.3.3 Port pins

In the flash memory programming mode, all the pins not used for flash memory programming enter the same status as that immediately after reset. If an external device connected to the ports does not recognize the port status immediately after reset, the port pin must be connected to either to V_{DD} or V_{SS} via a resistor.

17.3.4 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.

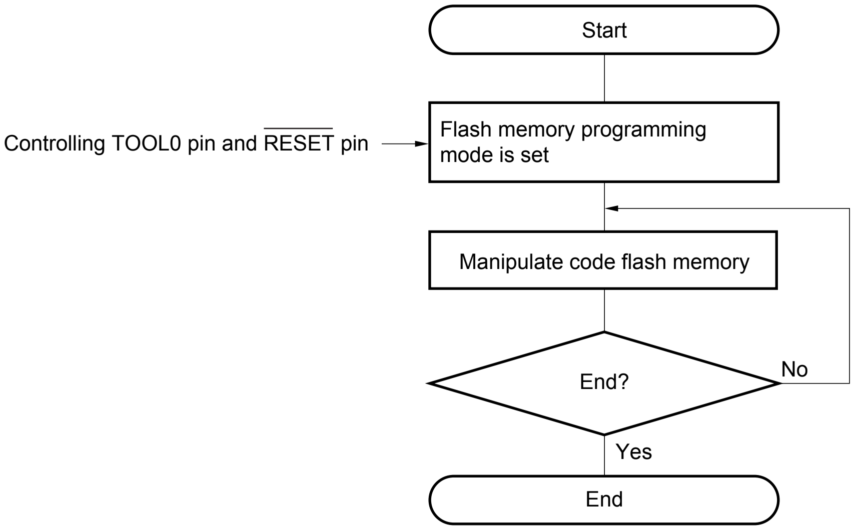
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

17.4 Serial Programming Method

17.4.1 Serial programming procedure

The following figure illustrates the procedure to rewrite the contents of the code flash memory by serial programming.

Figure 17-6. Code Flash Memory Manipulation Procedure



Refer to flash memory programming mode, see **17.4.2**.

<R> 17.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory by serial programming, the flash memory programming mode must be entered.

<When performing serial programming by using the dedicated flash memory programmer>

Connect the RL78 microcontroller to the dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode. The operating voltage during the flash memory programming mode is 4.5 V to 5.5 V.

<When performing serial programming by using an external device (UART communication)>

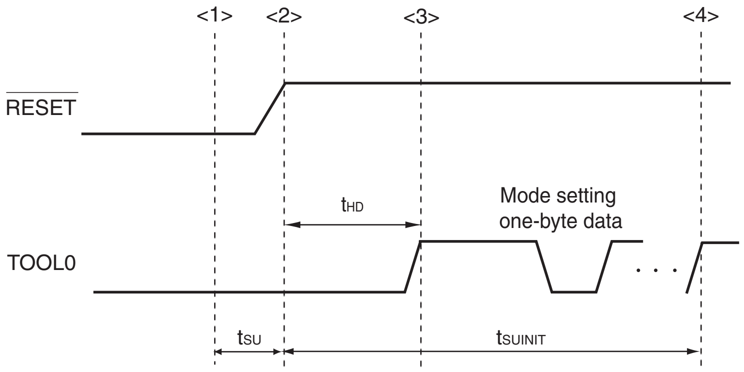
Set the TOOL0 pin to the low level, and then cancel the reset (refer to **Table 17-4**). The flash memory programming mode is then entered by following the procedures <1> to <4> shown in **Figure 17-7**.

The operating voltage during the flash memory programming mode is 4.5 V to 5.5 V.

Table 17-4. Relationship Between TOOL0 Pin and Operation Mode After Reset Release

TOOL0	Operation Mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

Figure 17-7. Entry to Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (SPOR reset must end before the external reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Setting of entry to the flash memory programming mode by UART reception.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the resets end.

t_{SU}: How long from when the TOOL0 pin is placed at the low level until an external reset ends

t_{HD}: How long to keep the TOOL0 pin at the low level from when the external reset ends

For details, refer to **21.9 Timing of Entry to Flash Memory Programming Modes**.

17.4.3 Communication mode

Communication mode of the RL78 microcontroller is as follows.

Table 17-5. Communication Mode

Communication Mode	Standard Setting[Note 1]				Pin Used
	Port	Speed [Note 2]	Frequency	Multiply Rate	
1-line mode	UART	115200 bps	–	–	TOOL0

Notes 1. Selection items for standard settings on GUI of the flash memory programmer.

2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

17.4.4 Communication commands

The RL78 microcontroller performs serial programming by using commands described in Table 17-6.

The signals sent from the flash memory programmer or external device to the RL78 microcontroller are called commands, and the RL78 microcontroller performs processing corresponding to the respective commands.

Table 17-6. Flash Memory Control Commands

Classification	Command Name	Function
CRC check	CRC check	Calculate checksums
Write after erase	Write after erase	Write data after erasing data in the flash memory

<R> 17.5 Processing Time of Each Command When Using PG-FP5 (Reference Values)

The processing time of each command (reference values) when using PG-FP5 as the dedicated flash memory programmer is shown below.

Table 17-7. Processing Time of Each Command When Using PG-FP5 (Reference Values)

Command of PG-FP5	Code Flash	
	1 KB	2 KB
	R7F0C80112ESP	R7F0C80212ESP
Write after erase	1.0 s	1.0 s
CRC check	0.5 s	0.5 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: 115,200 bps

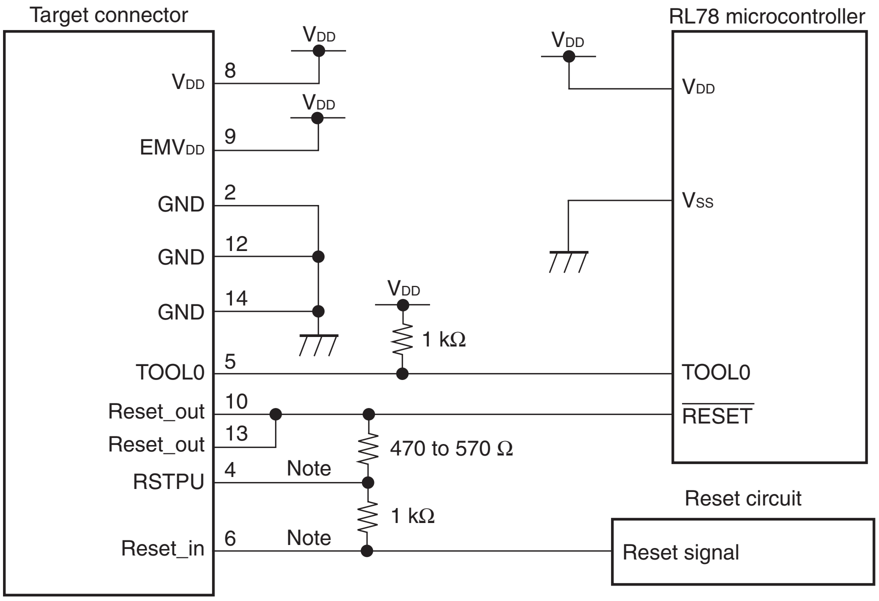
CHAPTER 18 ON-CHIP DEBUG FUNCTION

18.1 Connecting E1 On-chip Debugging Emulator

The RL78 microcontroller uses the V_{DD}, $\overline{RESET}$, TOOL0, and V_{SS} pins to communicate with the host machine via an E1 on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

Caution The RL78 microcontroller has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

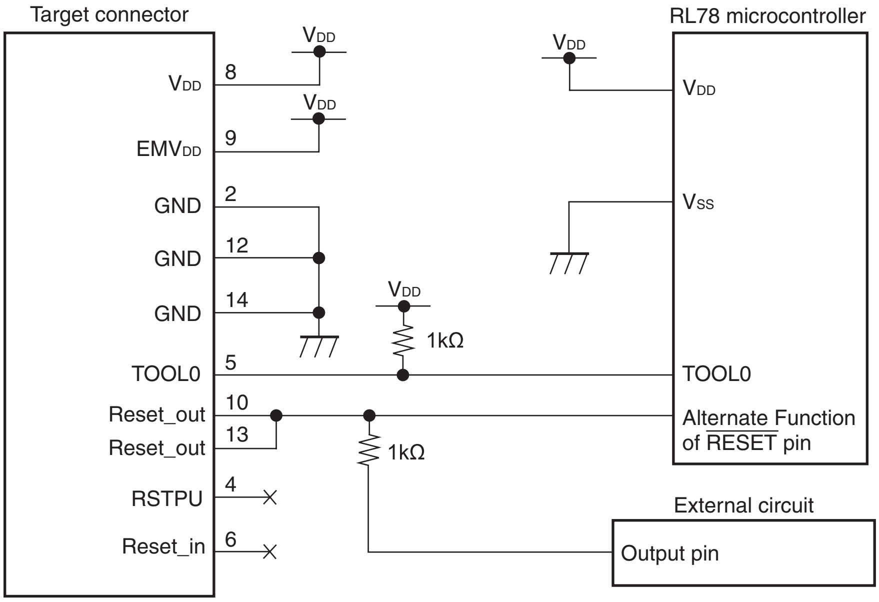
<R> **Figure 18-1. Connection Example of E1 On-chip Debugging Emulator and R7F0C80112ESP, R7F0C80212ESP**



Note Connecting the dotted line is not necessary during flash programming.

For the target system which uses the multi-use feature of $\overline{\text{RESET}}$ pin, its connection to an external circuit should be isolated.

<R> **Figure 18-2. Connection Example of E1 On-chip Debugging Emulator and R7F0C80112ESP, R7F0C80212ESP (When using to the alternative function of $\overline{\text{RESET}}$ pin)**



18.2 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 16 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

Table 18-1. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes

18.3 Securing of User Resources

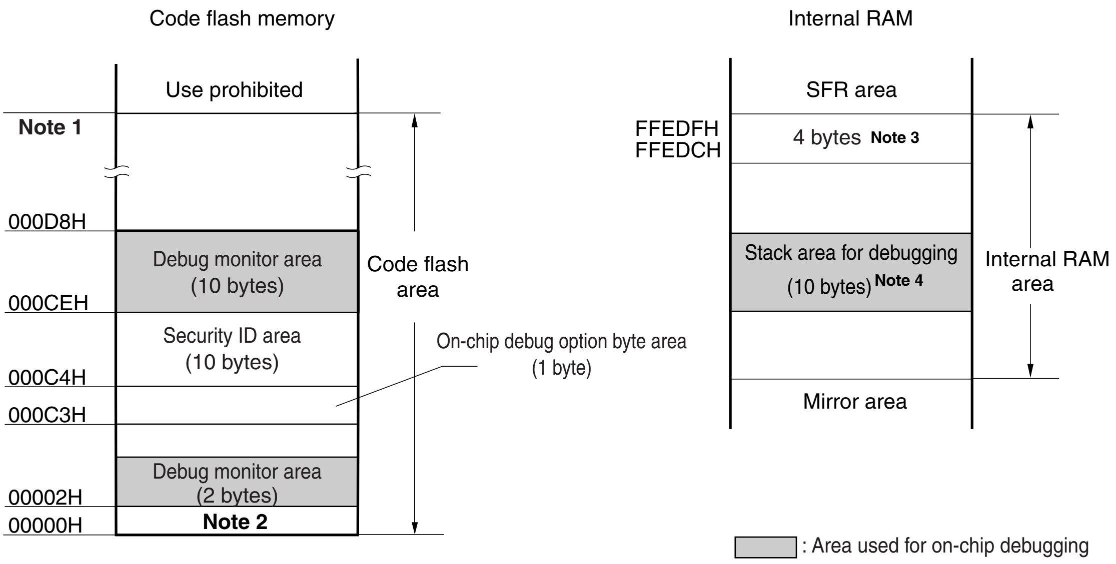
To perform communication between the RL78 microcontroller and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using linker options.

(1) Securement of memory space

The shaded portions in Figure 18-3 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.

<R> **Figure 18-3. Memory Spaces Where Debug Monitor Programs Are Allocated**



Notes 1. Address differs depending on products as follows.

Products	Address
R7F0C80112ESP	003FFH
R7F0C80212ESP	007FFH

2. In debugging, reset vector is rewritten to address allocated to a monitor program.

3. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are used, four bytes from FFEDCH to FFEDFH are consumed. Otherwise, this area can be used as the internal RAM.

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, 10 extra bytes are consumed for the stack area used.

CHAPTER 19 BCD CORRECTION CIRCUIT

19.1 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).

19.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

19.2.1 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 19-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

19.3 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 20 INSTRUCTION SET

This chapter lists the instructions for the RL78-S1 core of the RL78 microcontroller. For details of each operation and operation code, refer to the separate document **RL78 Microcontrollers User's Manual: software (R01US0015)**.

Remark The RL78-S2 core shares all instructions with the RL78-S1 core. Note, however, that the cores take different numbers of clock cycles to execute some instructions. The instructions which require different numbers of clock cycles are indicated by shading in the table under **20.2 Operation List**.

20.1 Conventions Used in Operation List

20.1.1 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 Table 20-1 below, R0, R1, R2, etc.) can be used for description.

Table 20-1. 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$^{\text{Note}}$) FFF00H to FFFFFH
saddr saddrp	FFE20H to FFF1FH Immediate data or labels FFE20H to FF1FH Immediate data or labels (even addresses only$^{\text{Note}}$)
addr20 addr16 addr5	00000H to FFFFFH Immediate data or labels 0000H to FFFFH Immediate data or labels (Automatically adds F to the top. only even addresses for 16-bit data transfer instructions$^{\text{Note}}$) 0080H to 00BFH Immediate data or labels (specification to bits 5 to 1, even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label

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 3-4 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 3-5 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

20.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 20-2. 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
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)
$\wedge$	Logical product (AND)
$\vee$	Logical sum (OR)
$\veebar$	Exclusive logical sum (exclusive OR)
$\overline{\ }$	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)

20.1.3 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 20-3. 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

20.1.4 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 20-4. 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.

20.2 Operation List

Table 20-5. 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 (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 program memory area is accessed.

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 20-5. 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 program memory area is accessed.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 20-5. 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 program memory area is accessed.

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.

<R>

Table 20-5. 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	2	–	rp ← word			
		saddrp, #word	4	2	–	(saddrp) ← word			
		sfrp, #word	4	2	–	sfrp ← word			
		AX, rp Note 3	1	2	–	AX ← rp			
		rp, AX Note 3	1	2	–	rp ← AX			
		AX, !addr16	3	2	5	AX ← (addr16)			
		!addr16, AX	3	2	–	(addr16) ← AX			
		AX, ES:!addr16	4	3	6	AX ← (ES, addr16)			
		ES:!addr16, AX	4	3	–	(ES, addr16) ← AX			
		AX, saddrp	2	2	–	AX ← (saddrp)			
		saddrp, AX	2	2	–	(saddrp) ← AX			
		AX, sfrp	2	2	–	AX ← sfrp			
		sfrp, AX	2	2	–	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 program memory area is accessed.

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.

<R>

Table 20-5. 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	2	5	AX ← (DE)			
		[DE], AX	1	2	–	(DE) ← AX			
		AX, ES:[DE]	2	3	6	AX ← (ES, DE)			
		ES:[DE], AX	2	3	–	(ES, DE) ← AX			
		AX, [HL]	1	2	5	AX ← (HL)			
		[HL], AX	1	2	–	(HL) ← AX			
		AX, ES:[HL]	2	3	6	AX ← (ES, HL)			
		ES:[HL], AX	2	3	–	(ES, HL) ← AX			
		AX, [DE+byte]	2	2	5	AX ← (DE+byte)			
		[DE+byte], AX	2	2	–	(DE+byte) ← AX			
		AX, ES:[DE+byte]	3	3	6	AX ← ((ES, DE) + byte)			
		ES:[DE+byte], AX	3	3	–	((ES, DE) + byte) ← AX			
		AX, [HL+byte]	2	2	5	AX ← (HL + byte)			
		[HL+byte], AX	2	2	–	(HL + byte) ← AX			
		AX, ES:[HL+byte]	3	3	6	AX ← ((ES, HL) + byte)			
		ES:[HL+byte], AX	3	3	–	((ES, HL) + byte) ← AX			
		AX, [SP+byte]	2	2	–	AX ← (SP + byte)			
		[SP+byte], AX	2	2	–	(SP + byte) ← AX			
		AX, word[B]	3	2	5	AX ← (B + word)			
		word[B], AX	3	2	–	(B+ word) ← AX			
		AX, ES:word[B]	4	3	6	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	3	–	((ES, B) + word) ← AX			
		AX, word[C]	3	2	5	AX ← (C + word)			
		word[C], AX	3	2	–	(C + word) ← AX			
		AX, ES:word[C]	4	3	6	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	3	–	((ES, C) + word) ← AX			
		AX, word[BC]	3	2	5	AX ← (BC + word)			
		word[BC], AX	3	2	–	(BC + word) ← AX			
		AX, ES:word[BC]	4	3	6	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	3	–	((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 program memory area is accessed.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

<R>

Table 20-5. 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	2	5	BC ← (addr16)			
		BC, ES:!addr16	4	3	6	BC ← (ES, addr16)			
		DE, !addr16	3	2	5	DE ← (addr16)			
		DE, ES:!addr16	4	3	6	DE ← (ES, addr16)			
		HL, !addr16	3	2	5	HL ← (addr16)			
		HL, ES:!addr16	4	3	6	HL ← (ES, addr16)			
		BC, saddrp	2	2	–	BC ← (saddrp)			
		DE, saddrp	2	2	–	DE ← (saddrp)			
		HL, saddrp	2	2	–	HL ← (saddrp)			
	XCHW	AX, rp Note 3	1	2	–	AX ←→ rp			
	ONEW	AX	1	2	–	AX ← 0001H			
		BC	1	2	–	BC ← 0001H			
	CLRW	AX	1	2	–	AX ← 0000H			
		BC	1	2	–	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 (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 program memory area is accessed.

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 20-5. 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 program memory area is accessed.

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 20-5. 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 program memory area is accessed.

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 20-5. 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 program memory area is accessed.

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 20-5. 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 program memory area is accessed.

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 20-5. 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	2	–	AX, CY ← AX+word	×	×	×
		AX, AX	1	2	–	AX, CY ← AX+AX	×	×	×
		AX, BC	1	2	–	AX, CY ← AX+BC	×	×	×
		AX, DE	1	2	–	AX, CY ← AX+DE	×	×	×
		AX, HL	1	2	–	AX, CY ← AX+HL	×	×	×
		AX, !addr16	3	2	5	AX, CY ← AX+(addr16)	×	×	×
		AX, ES:!addr16	4	3	6	AX, CY ← AX+(ES:addr16)	×	×	×
		AX, saddrp	2	2	–	AX, CY ← AX+(saddrp)	×	×	×
		AX, [HL+byte]	3	2	5	AX, CY ← AX+(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	3	6	AX, CY ← AX+((ES:HL)+byte)	×	×	×
	SUBW	AX, #word	3	2	–	AX, CY ← AX – word	×	×	×
		AX, BC	1	2	–	AX, CY ← AX – BC	×	×	×
		AX, DE	1	2	–	AX, CY ← AX – DE	×	×	×
		AX, HL	1	2	–	AX, CY ← AX – HL	×	×	×
		AX, !addr16	3	2	5	AX, CY ← AX – (addr16)	×	×	×
		AX, ES:!addr16	4	3	6	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, saddrp	2	2	–	AX, CY ← AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	2	5	AX, CY ← AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	3	6	AX, CY ← AX – ((ES:HL)+byte)	×	×	×
	CMPW	AX, #word	3	2	–	AX – word	×	×	×
		AX, BC	1	2	–	AX – BC	×	×	×
		AX, DE	1	2	–	AX – DE	×	×	×
		AX, HL	1	2	–	AX – HL	×	×	×
		AX, !addr16	3	2	5	AX – (addr16)	×	×	×
		AX, ES:!addr16	4	3	6	AX – (ES:addr16)	×	×	×
		AX, saddrp	2	2	–	AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	2	5	AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	3	6	AX – ((ES:HL)+byte)	×	×	×
Multiply	MULU	X	1	2	–	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 program memory area is accessed.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 20-5. 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	2	–	rp ← rp+1			
		!addr16	3	4	–	(addr16) ← (addr16)+1			
		ES:!addr16	4	5	–	(ES, addr16) ← (ES, addr16)+1			
		saddrp	2	4	–	(saddrp) ← (saddrp)+1			
		[HL+byte]	3	4	–	(HL+byte) ← (HL+byte)+1			
		ES: [HL+byte]	4	5	–	((ES:HL)+byte) ← ((ES:HL)+byte)+1			
	DECW	rp	1	2	–	rp ← rp – 1			
		!addr16	3	4	–	(addr16) ← (addr16) – 1			
		ES:!addr16	4	5	–	(ES, addr16) ← (ES, addr16) – 1			
		saddrp	2	4	–	(saddrp) ← (saddrp) – 1			
		[HL+byte]	3	4	–	(HL+byte) ← (HL+byte) – 1			
		ES: [HL+byte]	4	5	–	((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	2	–	(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	2	–	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) ×cnt			×
		BC, cnt	2	2	–	(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	2	–	(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 program memory area is accessed.

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 20-5. 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	2	–	(CY ← AX_{15}, AX_0 ← CY, AX_{m+1} ← AX_m) ×1			×
		BC,1	2	2	–	(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	–	CYX ← 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 program memory area is accessed.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 20-5. 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 program memory area is accessed.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 20-5. Operation List (15/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
Call/ return	CALL	rp	2	4	–	(SP – 2) ← (PC+2)$_S$, (SP – 3) ← (PC+2)$_H$, (SP – 4) ← (PC+2)$_L$, PC ← CS, rp, SP ← SP – 4			
		$!addr20	3	4	–	(SP – 2) ← (PC+3)$_S$, (SP – 3) ← (PC+3)$_H$, (SP – 4) ← (PC+3)$_L$, PC ← PC+3+jdisp16, SP ← SP – 4			
		!addr16	3	4	–	(SP – 2) ← (PC+3)$_S$, (SP – 3) ← (PC+3)$_H$, (SP – 4) ← (PC+3)$_L$, PC ← 0000, addr16, SP ← SP – 4			
		!!addr20	4	4	–	(SP – 2) ← (PC+4)$_S$, (SP – 3) ← (PC+4)$_H$, (SP – 4) ← (PC+4)$_L$, PC ← addr20, SP ← SP – 4			
	CALLT	[addr5]	2	6	–	(SP – 2) ← (PC+2)$_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	7	–	(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	7	–	PC$_L$ ← (SP), PC$_H$ ← (SP+1), PC$_S$ ← (SP+2), SP ← SP+4			
	RETI	-	2	8	–	PC$_L$ ← (SP), PC$_H$ ← (SP+1), PC$_S$ ← (SP+2), PSW← (SP+3), SP ← SP+4	R	R	R
	RETB	-	2	8	–	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 program memory area is accessed.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 20-5. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulate	PUSH	PSW	2	2	–	(SP – 1) ← PSW, (SP – 2) ← 00H, SP ← SP–2			
		rp	1	2	–	(SP – 1) ← rp$_H$, (SP – 2) ← rp$_L$, SP ← SP – 2			
	POP	PSW	2	4	–	PSW ← (SP+1), SP ← SP + 2	R	R	R
		rp	1	2	–	rp$_L$ ←(SP), rp$_H$ ← (SP+1), SP ← SP + 2			
	MOVW	SP, #word	4	2	–	SP ← word			
		SP, AX	2	2	–	SP ← AX			
		AX, SP	2	2	–	AX ← SP			
		HL, SP	3	2	–	HL ← SP			
		BC, SP	3	2	–	BC ← SP			
		DE, SP	3	2	–	DE ← SP			
	ADDW	SP, #byte	2	2	–	SP ← SP + byte			
	SUBW	SP, #byte	2	2	–	SP ← SP – byte			
Unconditional branch	BR	AX	2	3	–	PC ← CS, AX			
		$addr20	2	3	–	PC ← PC + 2 + jdisp8			
		$!addr20	3	3	–	PC ← PC + 3 + jdisp16			
		!addr16	3	3	–	PC ← 0000, addr16			
		!!addr20	4	3	–	PC ← addr20			
Conditional branch	BC	$addr20	2	2/4 $^{\text{Note3}}$	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4 $^{\text{Note3}}$	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4 $^{\text{Note3}}$	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4 $^{\text{Note3}}$	–	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4 $^{\text{Note3}}$	–	PC ← PC + 3 + jdisp8 if (Z∨CY)=0			
	BNH	$addr20	3	2/4 $^{\text{Note3}}$	–	PC ← PC + 3 + jdisp8 if (Z∨CY)=1			
	BT	saddr.bit, $addr20	4	3/5 $^{\text{Note3}}$	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5 $^{\text{Note3}}$	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5 $^{\text{Note3}}$	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5 $^{\text{Note3}}$	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5 $^{\text{Note3}}$	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6 $^{\text{Note3}}$	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 program memory area is accessed.

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 20-5. 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	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 program memory area is accessed.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

CHAPTER 21 ELECTRICAL SPECIFICATIONS

Cautions 1. The R7F0C80112ESP, R7F0C80212ESP has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

<R> **2. Use this product within the voltage range from 2.57 to 5.5 V because the detection voltage (V_{SPOR}) of the selectable power-on-reset (SPOR) circuit should also be considered.**

21.1 Absolute Maximum Ratings

(T_A = 25°C)

Parameter	Symbols	Conditions		Ratings	Unit
Supply Voltage	V_{DD}			–0.5 to +6.5	V
Input Voltage	V_{I1}			–0.3 to V_{DD} + 0.3^{Note}	V
Output Voltage	V_{O1}			–0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH1}	Per pin		–40	mA
		Total of all pins –140 mA	P40	–40	mA
			P00 to P04	–100	mA
Output current, low	I_{OL1}	Per pin		40	mA
		Total of all pins 140 mA	P40	40	mA
			P00 to P04	100	mA
Operating ambient temperature	T_A			–40 to +85	°C
Storage temperature	T_{stg}			–65 to +150	°C

Note 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.

Remarks
1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.
2. The reference voltage is V_{SS}.

21.2 Oscillator Characteristics

21.2.1 On-chip oscillator characteristics

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
<R> High-speed on-chip oscillator oscillation frequency[Note 1, 2]	f_{IH}		1.25		20	MHz
High-speed on-chip oscillator oscillation frequency accuracy		T_A = –40 to +85°C	–3		+3	%
		T_A = 0 to +40°C	–2		+2	%
Low-speed on-chip oscillator oscillation frequency [Note 3]	f_{IL}			15		kHz

Notes **1.** High-speed on-chip oscillator frequency is selected by bits 0 to 2 of option byte (000C2H).

2. This only indicates the oscillator characteristics. Refer to AC Characteristics for instruction execution time.

3. This only indicates the oscillator characteristics.

21.3 DC Characteristics

21.3.1 Pin characteristics

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

	Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
	Output current, high	I_{OH1}	P00[Note 1], P01, P02 to P04, P40	Per pin				–10.0[Note 3]	mA
			P40	Total[Note 2]	4.0 V ≤ V_{DD} ≤ 5.5 V			–10.0	mA
<R>					2.7 V ≤ V_{DD} < 4.0 V			–2.0	mA
<R>					2.4 V ≤ V_{DD} < 2.7 V			–1.5	mA
			P00[Note 1], P01, P02 to P04	Total[Note 2]	4.0 V ≤ V_{DD} ≤ 5.5 V			–50.0	mA
<R>					2.7 V ≤ V_{DD} < 4.0 V			–10.0	mA
<R>					2.4 V ≤ V_{DD} < 2.7 V			–7.5	mA
			Total of all pins[Note 2]					–60.0	mA
	Output current, low	I_{OL1}	P00 to P04, P40	Per pin				20.0[Note 3]	mA
			P40	Total[Note 2]	4.0 V ≤ V_{DD} ≤ 5.5 V			20.0	mA
<R>					2.7 V ≤ V_{DD} < 4.0 V			3.0	mA
<R>					2.4 V ≤ V_{DD} < 2.7 V			0.6	mA
			P00 to P04	Total[Note 2]	4.0 V ≤ V_{DD} ≤ 5.5 V			80.0	mA
<R>					2.7 V ≤ V_{DD} < 4.0 V			12.0	mA
<R>					2.4 V ≤ V_{DD} < 2.7 V			2.4	mA
			Total of all pins[Note 2]					100.0	mA
	Input voltage, high	V_{IH1}				0.8 V_{DD}		V_{DD}[Note 4]	V
	Input voltage, low	V_{IL1}				0		0.2 V_{DD}	V
	Output voltage, high	V_{OH1}	4.0 V ≤ V_{DD} ≤ 5.5 V		I_{OH} = –10 mA	V_{DD}–1.5			V
					I_{OH} = –3.0 mA	V_{DD}–0.7			V
<R>			2.7 V ≤ V_{DD} ≤ 5.5 V		I_{OH} = –2.0 mA	V_{DD}–0.6			V
<R>			2.4 V ≤ V_{DD} ≤ 5.5 V		I_{OH} = –1.5 mA	V_{DD}–0.5			V
	Output voltage, low	V_{OL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		I_{OH} = 20 mA			1.3	V
					I_{OH} = 8.5 mA			0.7	V
<R>			2.7 V ≤ V_{DD} ≤ 5.5 V		I_{OH} = 3.0 mA			0.6	V
					I_{OH} = 1.5 mA			0.4	V
<R>			2.4 V ≤ V_{DD} ≤ 5.5 V		I_{OH} = 0.6 mA			0.4	V
	Input leakage current, high	I_{LIH1}	V_I = V_{DD}					1	μA
	Input leakage current,low	I_{LIL1}	V_I = V_{SS}					–1	μA
	On-chip pull-up resistance	RU	V_I = V_{SS}			10	20	100	kΩ

Notes **1.** This pin does not output a high level in N-ch open-drain (V_{DD} tolerant) mode.

2. This is the output current value under conditions where the duty factor ≤ 70%.

The output current value when the duty factor > 70% can be calculated with the following expression (when changing the duty factor to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)
  <Example> Where n = 80 % and I_{OH} = - 10.0 mA
  Total output current of pins = (- 10.0 × 0.7)/(80 × 0.01) ≅ - 8.7 mA
- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)
  <Example> Where n = 80 % and I_{OL} = 10.0 mA
  Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≅ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

3. Do not exceed the total current value.

4. The maximum value of V_{IH} is V_{DD}, even in the N-ch open-drain (V_{DD} tolerant) mode.

Cautions 1. P00 does not output high level in N-ch open-drain mode.

2. The maximum value of V_{IH} of P00 is V_{DD} even in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port.

21.3.2 Supply current characteristics

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current$^{\text{Note 1}}$	I_{DD1}	Operating mode	Basic operation	f_{IH} = 20 MHz	V_{DD} = 3.0 V, 5.0 V		0.91		mA
			Normal operation	f_{IH} = 20 MHz	V_{DD} = 3.0 V, 5.0 V		1.57	2.04	
				f_{IH} = 5 MHz	V_{DD} = 3.0 V, 5.0 V		0.85	1.15	
	$I_{DD2}$$^{\text{Note 2}}$	HALT mode		f_{IH} = 20 MHz	V_{DD} = 3.0 V, 5.0 V		350	820	μA
				f_{IH} = 5 MHz	V_{DD} = 3.0 V, 5.0 V		290	600	
	$I_{DD3}$$^{\text{Note 3}}$	STOP mode		V_{DD} = 3 V			0.56	2.00	μA
WDT supply current$^{\text{Note 4}}$	I_{WDT}	f_{IL} = 15 kHz					0.31		μA
ADC supply current$^{\text{Note 5}}$	I_{ADC}	During conversion at the highest speed		V_{DD} = 5.0 V			1.30	1.90	mA
				V_{DD} = 3.0 V			0.50		

Notes
1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, I/O port, and on-chip pull-up/pull-down resistors.
2. During HALT instruction execution by flash memory.
3. When watchdog timer and A/D converter are stopped. The values below the MAX. column include the leakage current.
4. Current flowing only to the watchdog timer (including the operating current of the 15-kHz low-speed on-chip oscillator). The supply current value of the RL78 microcontroller is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates.
5. Current flowing only to the A/D converter. The supply current value of the RL78 microcontroller is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

Remarks
1. f_{IL}: Low-speed on-chip oscillator clock frequency
2. f_{IH}: High-speed on-chip oscillator clock frequency
3. Temperature condition of the TYP. value is T_A = 25°C

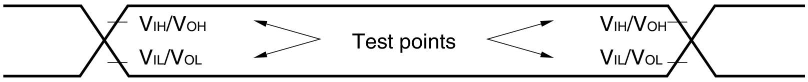
21.4 AC Characteristics

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

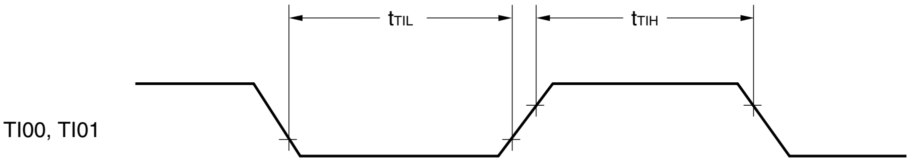
Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	2.7 V ≤ V_{DD} ≤ 5.5 V	0.05		0.8	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	0.2		0.8	μs
TI00, TI01 input high-level width, low-level width	t_{TIH}, t_{TIL}	Noise filter is not used		1/f_{MCK}+10			ns
TO00, TO01 output frequency	f_{TO}	4.0 V ≤ V_{DD} ≤ 5.5 V				10	MHz
		2.7 V ≤ V_{DD} < 4.0 V				5	MHz
		2.4 V ≤ V_{DD} < 2.7 V				2.5	MHz
PCLBUZ0 output frequency	f_{PCL}	4.0 V ≤ V_{DD} ≤ 5.5 V				10	MHz
		2.7 V ≤ V_{DD} < 4.0 V				5	MHz
		2.4 V ≤ V_{DD} < 2.7 V				2.5	MHz
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			μs

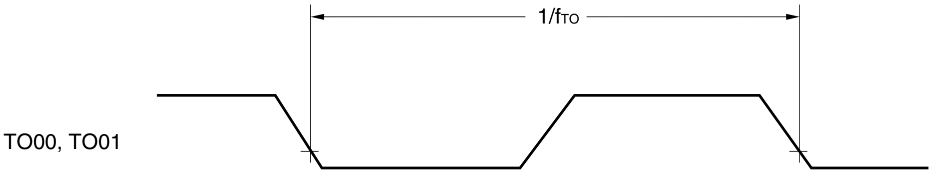
Remark f_{MCK}: Timer array unit operation clock frequency

AC Timing Test Points

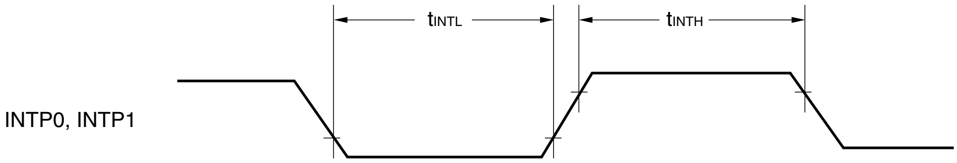


TI/TO Timing

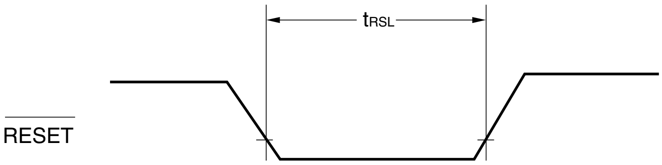




Interrupt Request Input Timing



$\overline{\text{RESET}}$ Input Timing



21.5 Serial Interface Characteristics

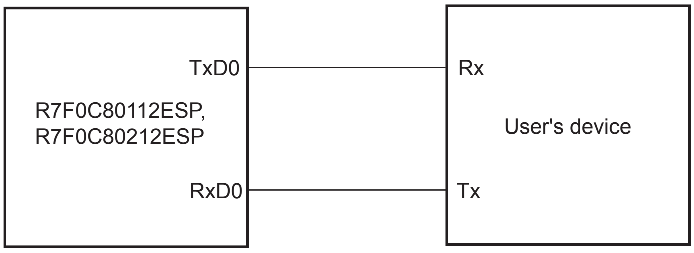
21.5.1 Serial array unit

(1) UART mode (dedicated baud rate generator output)

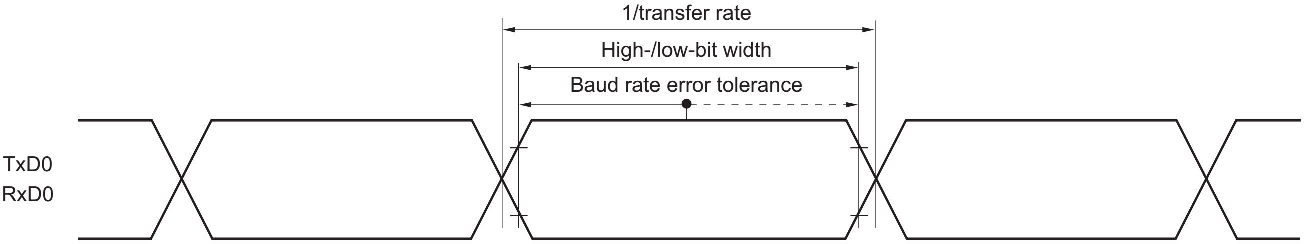
(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					$f_{MCK}/6$	bps
		Theoretical value of the maximum transfer rate $f_{CLK} = f_{MCK}$ = 20 MHz			3.3	Mbps

UART mode connection diagram



UART mode bit width (reference)



Caution **Select the normal output mode for the TxD0 pin by using port output mode register 0 (POM0).**

Remark f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00))

(2) CSI mode (master mode, SCKp...internal clock output)

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t_{KCY1}	2.7 V ≤ V_{DD} ≤ 5.5 V	200Note 1			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V	800Note 1			ns
$\overline{\text{SCKp}}$ high-/low-level width	t_{KH1}, t_{KL1}	2.7 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY1}/2 - 18$			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V	$t_{KCY1}/2 - 50$			ns
SIp setup time (to $\overline{\text{SCKp}}$↑)Note 2	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 5.5 V	47			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V	110			ns
SIp hold time (from SCKp↑)Note 3	t_{KSI1}		19			ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp outputNote 4	t_{KSO1}	C = 30 pFNote 5			25	ns

Notes 1. The value must also be 4/f_{CLK} or more.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{\text{SCKp}}$↓" 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 $\overline{\text{SCKp}}$↓" 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 $\overline{\text{SCKp}}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

5. C is the load capacitance of the $\overline{\text{SCKp}}$ and SOp output lines.

Caution **Select the the normal output mode for the SOp pin and $\overline{\text{SCKp}}$ pin by using port output mode register 0 (POM0).**

Remark p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0)

(3) CSI mode (slave mode, SCKp…external clock input)

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 5.5 V	f_{MCK} = 20 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 20 MHz	6/f_{MCK}			ns
		2.4 V ≤ V_{DD} < 2.7 V		6/f_{MCK}			ns
$\overline{\text{SCKp}}$ high-/low-level width	t_{KH2}, t_{KL2}	2.4 V ≤ V_{DD} ≤ 5.5 V		t_{KCY}/2			ns
SIp setup time (to $\overline{\text{SCKp}}$↑)$^{\text{Note 1}}$	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK}+20			ns
		2.4 V ≤ V_{DD} < 2.7 V		1/f_{MCK}+30			ns
SIp hold time (from $\overline{\text{SCKp}}$↑)$^{\text{Note 2}}$	t_{KSI2}	2.4 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK}+31			ns
Delay time from $\overline{\text{SCKp}}$↓ to SOp output$^{\text{Note 3}}$	t_{KSO2}	C = 30 pF$^{\text{Note 4}}$	2.7 V ≤ V_{DD} ≤ 5.5 V			2/f_{MCK}+50	ns
			2.4 V ≤ V_{DD} < 2.7 V			2/f_{MCK}+110	ns

Notes

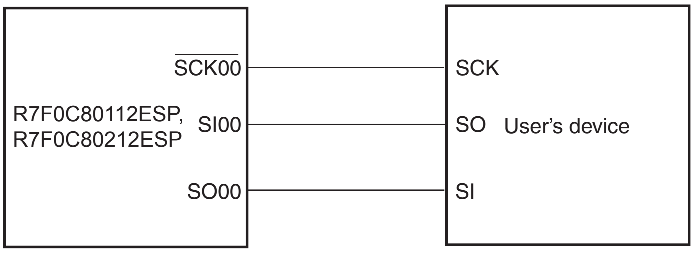
1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{\text{SCKp}}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.
2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{\text{SCKp}}$↓" when 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 $\overline{\text{SCKp}}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.
4. C is the load capacitance of the SOp output lines.

Caution Select the the normal output mode for the SOp pin by using port output mode register 0 (POM0).

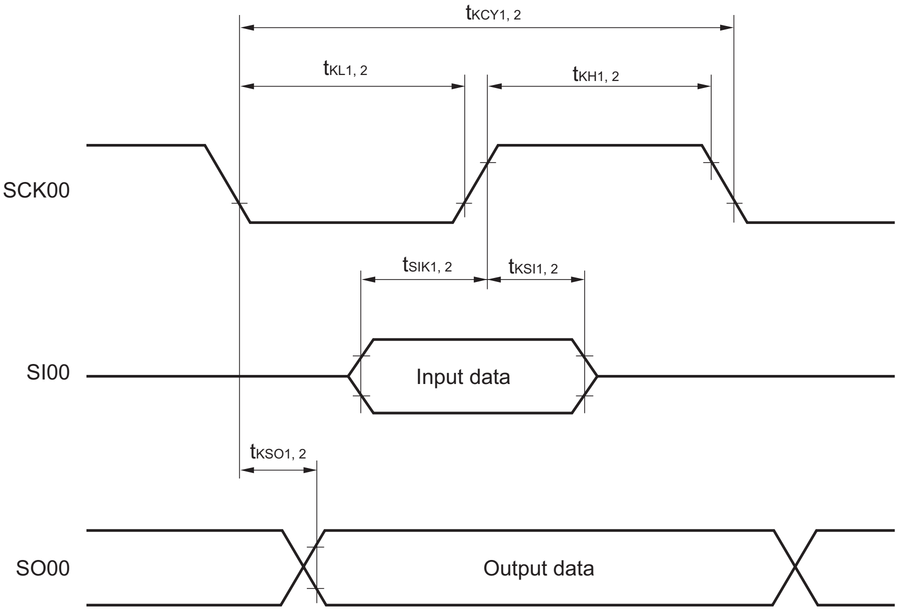
Remarks

1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0)
2. f_{MCK}: Serial array unit operation clock frequency
   (Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))

CSI mode connection diagram



CSI mode serial transfer timing
(When DAP00 = 0 and CKP00 = 0, or DAP00 = 1 and CKP00 = 1.)



21.6 Analog Characteristics

21.6.1 A/D converter characteristics

(Target ANI pin : ANI0 to ANI3)

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution	V_{DD} = 5 V		±1.7	±3.1[Note 2]	LSB
			V_{DD} = 3 V		±2.3	±4.5[Note 2]	LSB
Conversion time	t_{CONV}		2.7 V ≤ V_{DD} ≤ 5.5 V	3.4		18.4	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	4.6		18.4	μs
<R> Zero-scale error[Note 1]	E_{ZS}	10-bit resolution	V_{DD} = 5 V			±0.19[Note 2]	%FSR
<R>			V_{DD} = 3 V			±0.39[Note 2]	%FSR
<R> Full-scale error[Note 1]	E_{FS}	10-bit resolution	V_{DD} = 5 V			±0.29[Note 2]	%FSR
<R>			V_{DD} = 3 V			±0.42[Note 2]	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution	V_{DD} = 5 V			±1.8[Note 2]	LSB
			V_{DD} = 3 V			±1.7[Note 2]	LSB
Differential linearity error[Note 1]	DLE	10-bit resolution	V_{DD} = 5 V			±1.4[Note 2]	LSB
			V_{DD} = 3 V			±1.5[Note 2]	LSB
Analog input voltage	V_{AIN}			0		V_{DD}	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. MAX. value is the average value ±3σ at normalized distribution. Not tested in production.

21.6.2 Data retention power supply voltage characteristics

(T_A = –40 to +85°C, Vss = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage range	V_{DDDR}		1.9		5.5	V

Caution Data is retained until the power supply voltage becomes under the minimum value of the data retention power supply voltage range. Note that data in the RAM and RESF registers might not be cleared even if the power supply voltage becomes under the minimum value of the data retention power supply voltage range.

21.6.3 SPOR circuit characteristics

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection supply voltage	V_{SPOR0}	Power supply rise time		4.28		V
		Power supply fall time	4.00			V
	V_{SPOR1}	Power supply rise time		2.90		V
		Power supply fall time	2.70			V
	V_{SPOR2}	Power supply rise time		2.57		V
		Power supply fall time	2.40			V
<R> Minimum pulse width Note	T_{LSPW}		300			μs
Detection delay time					300	μs

Note Time required for the reset operation by the SPOR when V_{DD} becomes under V_{SPDR}.

<R> 21.6.4 Power supply voltage rising slope characteristics

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

21.7 Flash Memory Programming Characteristics

(T_A = 0 to + 40°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Code flash memory rewritable times Notes 1, 2, 3	C_{erwr}	1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.	Retained for 20 years (Self/serial programming)	100			Times

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.

3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

<R> 21.8 Dedicated Flash Memory Programmer Communication (UART)

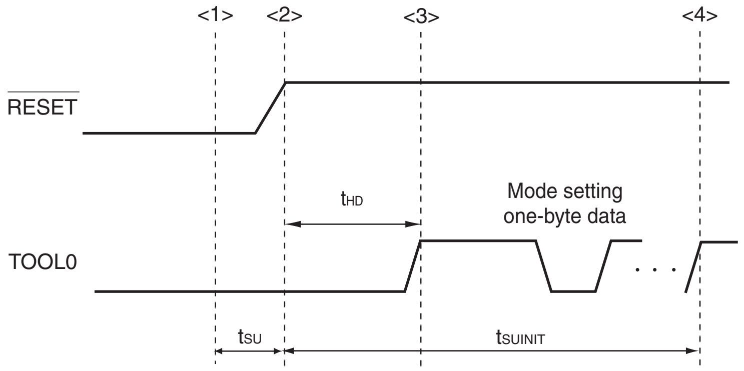
(T_A = - 40 to + 85 °C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate				115,200		bps

Remark The transfer rate during flash memory programming is fixed to 115,200 bps.

<R> 21.9 Timing of Entry to Flash Memory Programming Modes

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Time to complete the communication for the initial setting after the external reset is released	t_{SUINIT}	SPOR 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}	SPOR 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	t_{HD}	SPOR 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 (SPOR reset must be released before the external reset is released.).

<3> The TOOL0 pin is set to the high level.

<4> Setting of entry to the flash memory programming mode by UART reception.

Remark t_{SUINIT}: Communication for the initial setting must be completed within 100 ms after the external reset is released during this period.

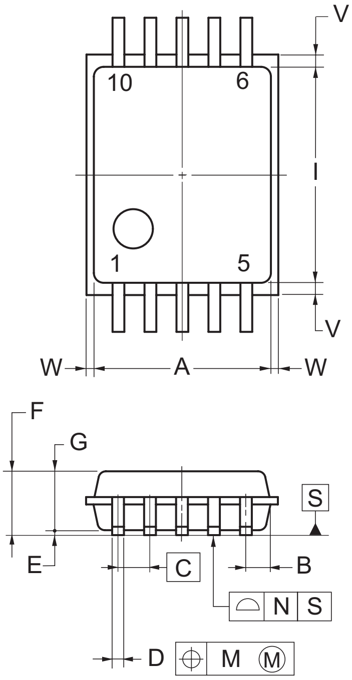
t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level

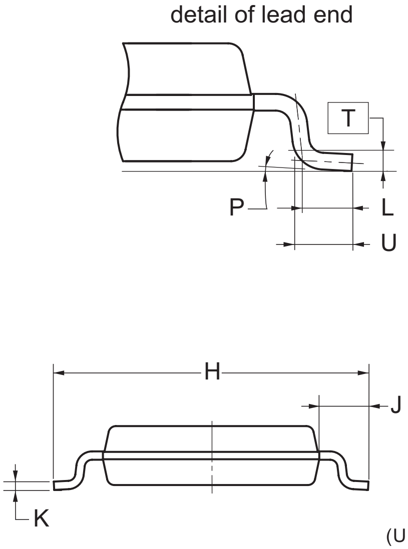
t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released

CHAPTER 22 PACKAGE DRAWINGS

R7F0C80112ESP, R7F0C80212ESP

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP10-4.4x3.6-0.65	PLSP0010JA-A	P10MA-65-CAC-2	0.05





NOTE
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

(UNIT:mm)

ITEM	DIMENSIONS
A	3.60±0.10
B	0.50
C	0.65 (T.P.)
D	0.24±0.08
E	0.10±0.05
F	1.45 MAX.
G	1.20±0.10
H	6.40±0.20
I	4.40±0.10
J	1.00±0.20
K	$0.17^{+0.08}_{-0.07}$
L	0.50
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25 (T.P.)
U	0.60±0.15
V	0.25 MAX.
W	0.15 MAX.

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

(1/3)

Page	Description	Classification
CHAPTER 1 OUTLINE		
p.1	Modification of **1.1 Features.**	(b)
p.2	Addition of **1.2 List of Part Numbers.**	(d)
p.5	Addition of Note to **1.6 Outline of Functions.**	(b), (c)
CHAPTER 2 PIN FUNCTIONS		
p. 7	Modification of **2.2 Functions other than port pins.**	(c)
CHAPTER 3 CPU ARCHITECTURE		
p. 10	Modification of **CHAPTER 3 CPU ARCHITECTURE.**	(c)
p. 11	Addition of Note to **Figure 3-1. Memory Map for the R7F0C80112ESP.**	(c)
p. 12	Addition of Note to **Figure 3-2. Memory Map for the R7F0C80212ESP.**	(c)
p. 17	Modification of **Figure 3-3. Correspondence Between Data Memory and Addressing.**	(c)
p. 19	Modification of **Figure 3-6. Format of Stack Pointer.**	(a)
p. 22	Modification of **3.2.3 ES and CS registers.**	(c)
p. 32	Modification of **3.4.3 Direct addressing.**	(c)
p. 35	Modification of **3.4.6 Register indirect addressing.**	(c)
p. 36	Modification of **3.4.7 Based addressing.**	(c)
p. 40	Modification of **3.4.8 Based indexed addressing.**	(c)
p. 41	Modification of **3.4.9 Stack addressing**	(c)
CHAPTER 4 PORT FUNCTIONS		
p. 44	Modification of **4.2.1 Port 0**	(c)
p. 53	Addition of Caution to **Figure 4-6. Format of Peripheral I/O Redirection Register (PIOR).**	(c)
pp.55, 56	Modification of **4.5 Register Settings When an Alternate Function Is Used**	(c)
pp. 57, 58	Modification of **Table 4-5. Examples of Register And Output Latch Settings With Pin Functions**	(c)
CHAPTER 5 CLOCK GENERATOR		
p. 61	Addition of Note to **5.1 Functions of Clock Generator.**	(c)
p. 66	Modification of Note to **Figure 5-3. Format of High-Speed On-Chip Oscillator Frequency Selection Register (HOCODIV).**	(b)
p. 68	Modification of Note to **Figure 5-4. Clock Generator Operation When Power Supply Voltage Is Turned On.**	(b)
p. 69	Modification of **5.6 Controlling Clock.**	(c)
p. 70	Modification of **Figure 5-5. CPU Clock Status Transition Diagram.**	(c)
CHAPTER 6 TIMER ARRAY UNIT		
p. 79	Addition of Caution to **6.2.1 Timer/counter register 0n (TCR0n).**	(c)
p. 80	Addition of Caution to **Table 6-2. Timer/counter Register 0n (TCR0n) Read Value in Various Operation Modes.**	(c)
p. 81	Modification of **6.2.2 Timer data register 0n (TDR0n).**	(c)
p. 163	Addition of Caution to **6.4.2 Basic rules of 8-bit timer operation function (only channel 1).**	(c)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(2/3)

Page	Description	Classification
p. 106	Modification of **Figure 6-24. Operation Timing (In Interval Timer Mode).**	(a)
p. 108	Modification of **Figure 6-26. Operation Timing (In Capture Mode: Input Pulse Interval Measurement).**	(a)
p. 109	Modification of **Figure 6-27. Operation Timing (In One-count Mode).**	(a)
p. 110	Modification of **Figure 6-28. Operation Timing (In Capture & One-count Mode: High-level Width Measurement).**	(a)
p. 124	Addition of Caution to **Figure 6-41. Operation Procedure of Interval Timer/Square Wave Output Function.**	(c)
p. 155	Addition of Caution to **6.8.2 Operation as PWM function.**	(c)
CHAPTER 7 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER		
p. 165	Modification of Caution to **Figure 7-2. Format of Clock Output Select Register 0 (CKS0).**	(c)
p. 167	Modification of **7.4.1 Operation as output pin.**	(c)
CHAPTER 9 A/D CONVERTER		
p. 173	Modification of **9.1 Function of A/D Converter**	(c)
p. 178	Modification of **9.3.2 A/D converter mode register 0 (ADM0).**	(c)
p. 181	Modification of **Figure 9-5. A/D Converter Sampling and A/D Conversion Timing**	(c)
p. 184	Modification of Caution to **9.3.6 Analog input channel specification register (ADS)**	(c)
p. 188	Modification of **Figure 9-13. Conversion Operation of A/D Converter**	(c)
p. 190	Modification of **9.6 A/D Converter Operation Modes**	(c)
p. 190	Modification of **Figure 9-15. Example of Operation Timing**	(c)
p. 191	Modification of **9.7 A/D Converter Setup Flowchart**	(c)
p. 194	Modification of **9.9.3 Conflicting operations**	(c)
p. 195	Modification of **Figure 9-22. Analog Input Pin Connection.**	(c)
p. 196	Modification of **Figure 9-23. Timing of A/D Conversion End Interrupt Request Generation.**	(c)
p. 197	Modification of **Table 9-4. Resistance and Capacitance Values of Equivalent Circuit.**	(c)
CHAPTER 10 SERIAL ARRAY UNIT		
p. 213	Modification of **10.3.5 Serial data register 0n (SDR0nH, SDR0nL).**	(c)
p. 228	Modification of Caution to **Figure 10-22. Peripheral Enable Register 0 (PER0) Setting When Stopping Operation by Units.**	(c)
p. 287	Modification of **10.5.7 Calculating transfer clock frequency**	(c)
CHAPTER 11 INTERRUPT FUNCTIONS		
p. 320	Modification of Caution to **Figure 11-5. Format of External Interrupt Rising Edge Enable Register (EGP0) and External Interrupt Falling Edge Enable Register (EGN0)**	(c)
CHAPTER 12 KEY INTERRUPT FUNCTION		
p. 330	Addition of Note to **Table 12-1. Assignment of Key Interrupt Detection Pins.**	(c)
p. 333	Modification of Caution to**12.3.2 Key return mode register (KRM0).**	(c)
p. 334	Addition of Caution to **12.3.3 Key return flag register (KRF).**	(c)
pp. 335 to 338	Addition of **12.4 Key Interrupt Operation.**	(c)
CHAPTER 13 STANDBY FUNCTION		
pp. 340, 341	Modification of **13.2.1 HALT mode.**	(c)
p. 343	Modification of Note, Remark to **Figure 13-3. STOP Mode Release by Interrupt Request Generation**	(b)

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

(3/3)

Page	Description	Classification
CHAPTER 14 RESET FUNCTION		
p. 345	Modification of Caution to **CHAPTER 14 RESET FUNCTION.**	(c)
p. 347	Modification of **Figure 14-2. Timing of Reset by $\overline{\text{RESET}}$ Input.**	(b)
p. 348	odification of Note to **Figure 14-3. Timing of Reset Due to Watchdog Timer Overflow or Execution of Illegal Instruction.**	(b)
CHAPTER 15 SELECTABLE POWER-ON-RESET CIRCUIT		
p. 355	Modification of **15.1 Functions of Selectable Power-on-reset Circuit.**	(c)
p. 356	Modification of **Figure 15-1. Block Diagram of Selectable Power-on-reset Circuit.**	(b)
p. 357	Modification of **15.3 Operation of Selectable Power-on-reset Circuit.**	(b)
p. 358	Modification of **15.4 Cautions for Selectable Power-on-reset Circuit.**	(b)
CHAPTER 16 OPTION BYTE		
p. 360	Modification of **Figure 16-1. Format of User Option Byte (000C0H).**	(c)
p. 361	Addition of Remark to **Figure 16-2. Format of User Option Byte (000C1H).**	(c)
CHAPTER 17 FLASH MEMORY		
p. 368	Modification of **17.2 Writing to Flash Memory by Using External Device (that Incorporates UART).**	(c)
p. 372	Modification of **17.4.2 Flash memory programming mode.**	(c)
p. 373	Modification of **17.5 Processing Time of Each Command When Using PG-FP5 (Reference Values).**	(c)
CHAPTER 18 ON-CHIP DEBUG FUNCTION		
p. 374	Modification of **Figure 18-1. Connection Example of E1 On-chip Debugging Emulator and R7F0C80112ESP, R7F0C80212ESP.**	(c)
p. 375	Modification of **Figure 18-2. Connection Example of E1 On-chip Debugging Emulator and R7F0C80112ESP, R7F0C80212ESP (When using to the alternative function of $\overline{\text{RESET}}$ pin).**	(c)
p. 377	Modification of **Figure 18-3. Memory Spaces Where Debug Monitor Programs Are Allocated.**	(c)
CHAPTER 20 INSTRUCTION SET		
pp. 388 to 390	Modification of **Table 20-5. Operation List.**	(c)
CHAPTER 21 ELECTRICAL SPECIFICATIONS		
p. 402	Addition of Caution to **CHAPTER 21 ELECTRICAL SPECIFICATIONS.**	(c)
p. 404	Modification of **21.2.1 On-chip oscillator characteristics.**	(a)
p. 405	Modification of **21.3.1 Pin characteristics.**	(a)
p. 414	Modification of **21.6.1 A/D converter characteristics.**	(b)
p. 415	Modification of **21.6.3 SPOR circuit characteristics.**	(c)
p. 415	Modification of **21.6.4 Power supply voltage rising slope characteristics.**	(b)
p. 416	Modification of **21.8 Dedicated Flash Memory Programmer Communication (UART).**	(b)
p. 417	Modification of **21.9 Timing of Entry to Flash Memory Programming Modes.**	(b)

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

R7F0C80112ESP, R7F0C80212ESP User’s Manual: Hardware

Publication Date: Rev.0.90 Jan 30, 2013
Rev.1.00 Sep 20, 2013

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.3

R7F0C80112ESP, R7F0C80212ESP

Renesas Electronics Corporation