

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. 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 categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - “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.
  - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
  - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. 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.
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.

# 78K0R/KH3

## 16-bit Single-Chip Microcontrollers

---

**$\mu$ PD78F1174, 78F1174A, 78F1174A(A)**

**$\mu$ PD78F1175, 78F1175A, 78F1175A(A)**

**$\mu$ PD78F1176, 78F1176A, 78F1176A(A)**

**$\mu$ PD78F1177, 78F1177A, 78F1177A(A)**

**$\mu$ PD78F1178, 78F1178A, 78F1178A(A)**

[MEMO]

## 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  $V_{IL}$  (MAX) and  $V_{IH}$  (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  $V_{IL}$  (MAX) and  $V_{IH}$  (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  $V_{DD}$  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.

**Windows and Windows NT are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.**

**PC/AT is a trademark of International Business Machines Corporation.**

**EEPROM is a trademark of NEC Electronics Corporation.**

**SuperFlash is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.**

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

- **The information in this document is current as of September, 2009. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**

- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.

- NEC 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 NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.

- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.

- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

"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.

"Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).

"Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

# INTRODUCTION

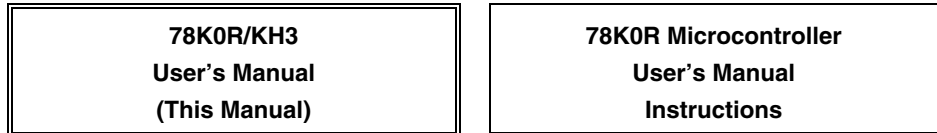
**Readers** This manual is intended for user engineers who wish to understand the functions of the 78K0R/KH3 and design and develop application systems and programs for these devices.

The target products are as follows.

- Conventional-specification products of the 78K0R/KH3:  
 $\mu$ PD78F1174, 78F1175, 78F1176, 78F1177, 78F1178
- Expanded-specification products of the 78K0R/KH3:  
 $\mu$ PD78F1174A, 78F1175A, 78F1176A, 78F1177A, 78F1178A
- (A) grade products of the expanded-specification products of the 78K0R/KH3 :  
 $\mu$ PD78F1174A(A), 78F1175A(A), 78F1176A(A), 78F1177A(A), 78F1178A(A)

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

**Organization** The 78K0R/KH3 manual is separated into two parts: this manual and the instructions edition (common to the 78K0R Microcontroller Series).



- |  |   |
|--|---|
| <ul style="list-style-type: none"><li>• Pin functions</li><li>• Internal block functions</li><li>• Interrupts</li><li>• Other on-chip peripheral functions</li><li>• Electrical specifications</li></ul> | <ul style="list-style-type: none"><li>• CPU functions</li><li>• Instruction set</li><li>• Explanation of each instruction</li></ul> |
|--|---|

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

- When using this manual as the manual for (A) grade products of the expanded-specification products of 78K0R/KH3 microcontrollers:
  - Only the electrical specifications and quality grade differ between standard products and (A) grade products. Read the part number for (A) grade products as follows.
    - $\mu$ PD78F117yA →  $\mu$ PD78F117yA(A) (y = 4 to 8)
- 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 RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R.
- To know details of the 78K0R Series instructions:
  - Refer to the separate document **78K0R Microcontroller Instructions User's Manual (U17792E)**.

### Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representations:	$\overline{xxx}$ (overscore over pin and signal name)
<b>Note:</b>	Footnote for item marked with <b>Note</b> in the text
<b>Caution:</b>	Information requiring particular attention
<b>Remark:</b>	Supplementary information
Numerical representations:	Binary        ...xxxx or xxxxB
	Decimal        ...xxxx
	Hexadecimal   ...xxxxH

### 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.
78K0R/KH3 User's Manual	This manual
78K0R Microcontroller Instructions User's Manual	U17792E
78K0R Microcontroller Self Programming Library Type01 User's Manual <sup>Note</sup>	U18706E

**Note** This document is under engineering management. For details, consult an NEC Electronics sales representative.

### Documents Related to Development Tools (Software) (User's Manuals)

Document Name	Document No.	
CC78K0R Ver. 2.00 C Compiler	Operation	U18549E
	Language	U18548E
RA78K0R Ver. 1.20 Assembler Package	Operation	U18547E
	Language	U18546E
SM+ System Simulator	Operation	U18601E
PM+ Ver. 6.30		U18416E
ID78K0R-QB Ver. 3.20 Integrated Debugger	Operation	U17839E

### Documents Related to Development Tools (Hardware) (User's Manuals)

Document Name	Document No.
QB-MINI2 On-Chip Debug Emulator with Programming Function	U18371E
QB-78K0RKX3 In-Circuit Emulator	U17866E



### Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP4 Flash Memory Programmer User's Manual	U15260E
PG-FP5 Flash Memory Programmer	U18865E

**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.
SEMICONDUCTOR SELECTION GUIDE – Products and Packages –	X13769X
Semiconductor Device Mount Manual	<b>Note</b>
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

**Note** See the “Semiconductor Device Mount Manual” website (<http://www.necel.com/pkg/en/mount/index.html>).

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

## CONTENTS

<b>CHAPTER 1 OUTLINE .....</b>	<b>18</b>
<b>1.1 Differences Between Conventional-Specification Products (<math>\mu</math>PD78F117x) and Expanded-Specification Products (<math>\mu</math>PD78F117xA).....</b>	<b>18</b>
<b>1.2 Features.....</b>	<b>20</b>
<b>1.3 Applications .....</b>	<b>21</b>
<b>1.4 Ordering Information.....</b>	<b>21</b>
<b>1.5 Pin Configuration (Top View) .....</b>	<b>22</b>
<b>1.6 78K0R/Kx3 Microcontroller Lineup.....</b>	<b>24</b>
<b>1.7 Block Diagram .....</b>	<b>25</b>
<b>1.8 Outline of Functions.....</b>	<b>26</b>
<b>CHAPTER 2 PIN FUNCTIONS .....</b>	<b>28</b>
<b>2.1 Pin Function List .....</b>	<b>28</b>
<b>2.2 Description of Pin Functions .....</b>	<b>36</b>
2.2.1 P00 to P07 (port 0).....	36
2.2.2 P10 to P17 (port 1).....	37
2.2.3 P20 to P27 (port 2).....	38
2.2.4 P30 to P37 (port 3).....	38
2.2.5 P40 to P47 (port 4).....	39
2.2.6 P50 to P57 (port 5).....	40
2.2.7 P60 to P67 (port 6).....	41
2.2.8 P70 to P77 (port 7).....	41
2.2.9 P80 to P87 (port 8).....	42
2.2.10 P90 to P97 (port 9).....	42
2.2.11 P110 to P117 (port 11).....	43
2.2.12 P120 to P127 (port 12).....	43
2.2.13 P130, P131 (port 13).....	44
2.2.14 P140 to P145 (port 14).....	45
2.2.15 P150 to P157 (port 15).....	46
2.2.16 P160 to P163 (port 16).....	46
2.2.17 AV <sub>REF0</sub> .....	47
2.2.18 AV <sub>REF1</sub> .....	48
2.2.19 AV <sub>SS</sub> .....	48
2.2.20 $\overline{\text{RESET}}$ .....	48
2.2.21 REGC.....	48
2.2.22 V <sub>DD</sub> , EV <sub>DD0</sub> , EV <sub>DD1</sub> .....	49
2.2.23 V <sub>SS</sub> , EV <sub>SS0</sub> , EV <sub>SS1</sub> .....	49
2.2.24 FLMD0 .....	49
<b>2.3 Pin I/O Circuits and Recommended Connection of Unused Pins .....</b>	<b>50</b>

<b>CHAPTER 3 CPU ARCHITECTURE</b> .....	<b>55</b>
<b>3.1 Memory Space</b> .....	<b>55</b>
3.1.1 Internal program memory space .....	63
3.1.2 Mirror area .....	65
3.1.3 Internal data memory space.....	66
3.1.4 Special function register (SFR) area .....	67
3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area .....	67
3.1.6 Data memory addressing .....	68
<b>3.2 Processor Registers</b> .....	<b>73</b>
3.2.1 Control registers.....	73
3.2.2 General-purpose registers .....	75
3.2.3 ES and CS registers .....	77
3.2.4 Special function registers (SFRs).....	78
3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers).....	84
<b>3.3 Instruction Address Addressing</b> .....	<b>91</b>
3.3.1 Relative addressing .....	91
3.3.2 Immediate addressing.....	91
3.3.3 Table indirect addressing .....	92
3.3.4 Register direct addressing .....	93
<b>3.4 Addressing for Processing Data Addresses</b> .....	<b>94</b>
3.4.1 Implied addressing.....	94
3.4.2 Register addressing .....	94
3.4.3 Direct addressing .....	95
3.4.4 Short direct addressing .....	96
3.4.5 SFR addressing .....	97
3.4.6 Register indirect addressing.....	98
3.4.7 Based addressing .....	99
3.4.8 Based indexed addressing.....	102
3.4.9 Stack addressing .....	103
<b>CHAPTER 4 PORT FUNCTIONS</b> .....	<b>104</b>
<b>4.1 Port Functions</b> .....	<b>104</b>
<b>4.2 Port Configuration</b> .....	<b>108</b>
4.2.1 Port 0 .....	109
4.2.2 Port 1 .....	117
4.2.3 Port 2 .....	123
4.2.4 Port 3 .....	125
4.2.5 Port 4 .....	128
4.2.6 Port 5 .....	137
4.2.7 Port 6 .....	139
4.2.8 Port 7 .....	142
4.2.9 Port 8 .....	143
4.2.10 Port 9 .....	144
4.2.11 Port 11 .....	148
4.2.12 Port 12 .....	150
4.2.13 Port 13 .....	155
4.2.14 Port 14 .....	157

4.2.15 Port 15.....	161
4.2.16 Port 16.....	162
<b>4.3 Registers Controlling Port Function .....</b>	<b>163</b>
<b>4.4 Port Function Operations .....</b>	<b>172</b>
4.4.1 Writing to I/O port .....	172
4.4.2 Reading from I/O port.....	172
4.4.3 Operations on I/O port.....	172
4.4.4 Connecting to external device with different potential (2.5 V, 3 V) .....	173
<b>4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function.....</b>	<b>175</b>
<b>4.6 Cautions on 1-Bit Manipulation Instruction for Port Register n (Pn).....</b>	<b>179</b>
<b>CHAPTER 5 EXTERNAL BUS INTERFACE .....</b>	<b>180</b>
<b>5.1 Functions of External Bus Interface.....</b>	<b>180</b>
<b>5.2 Registers Controlling External Bus Interface Functions .....</b>	<b>185</b>
<b>5.3 Setting Port Mode Register and Output Latch .....</b>	<b>188</b>
<b>5.4 Number of Instruction Wait Clocks or Data Access .....</b>	<b>189</b>
<b>5.5 Number of Instruction Execution Clocks and Instruction Wait Clocks for Fetch Access..</b>	<b>189</b>
<b>5.6 Number of Instructed Wait Cycles According to External Wait Cycles.....</b>	<b>190</b>
<b>5.7 Timing of External Bus Interface Function.....</b>	<b>191</b>
5.7.1 Multiplexed bus mode .....	192
5.7.2 Separate bus mode .....	196
<b>5.8 Example of Connection to Memory .....</b>	<b>200</b>
5.8.1 Connection of external logic (ASIC, etc.).....	200
5.8.2 Connection of synchronous memory .....	200
5.8.3 Connection of asynchronous memory .....	201
<b>CHAPTER 6 CLOCK GENERATOR .....</b>	<b>202</b>
<b>6.1 Functions of Clock Generator.....</b>	<b>202</b>
<b>6.2 Configuration of Clock Generator .....</b>	<b>203</b>
<b>6.3 Registers Controlling Clock Generator.....</b>	<b>205</b>
<b>6.4 System Clock Oscillator .....</b>	<b>219</b>
6.4.1 X1 oscillator.....	219
6.4.2 XT1 oscillator .....	219
6.4.3 Internal high-speed oscillator .....	222
6.4.4 Internal low-speed oscillator.....	222
6.4.5 Prescaler.....	222
<b>6.5 Clock Generator Operation .....</b>	<b>223</b>
<b>6.6 Controlling Clock.....</b>	<b>227</b>
6.6.1 Example of controlling high-speed system clock.....	227
6.6.2 Example of controlling internal high-speed oscillation clock.....	230
6.6.3 Example of controlling subsystem clock.....	232
6.6.4 Example of controlling internal low-speed oscillation clock .....	234
6.6.5 CPU clock status transition diagram.....	235
6.6.6 Condition before changing CPU clock and processing after changing CPU clock .....	240
6.6.7 Time required for switchover of CPU clock and main system clock .....	242
6.6.8 Conditions before clock oscillation is stopped .....	243

<b>CHAPTER 7 TIMER ARRAY UNIT .....</b>	<b>244</b>
<b>7.1 Functions of Timer Array Unit .....</b>	<b>244</b>
7.1.1 Functions of each channel when it operates independently.....	244
7.1.2 Functions of each channel when it operates with another channel .....	245
7.1.3 LIN-bus supporting function (channel 7 only).....	245
<b>7.2 Configuration of Timer Array Unit.....</b>	<b>246</b>
<b>7.3 Registers Controlling Timer Array Unit .....</b>	<b>252</b>
<b>7.4 Channel Output (TOmn pin) Control .....</b>	<b>276</b>
7.4.1 TOmn pin output circuit configuration .....	276
7.4.2 TOmn Pin Output Setting .....	277
7.4.3 Cautions on Channel Output Operation .....	278
7.4.4 Collective manipulation of TOmn bits.....	281
7.4.5 Timer Interrupt and TOmn Pin Output at Operation Start .....	282
<b>7.5 Channel Input (Tlmn Pin) Control.....</b>	<b>283</b>
7.5.1 Tlmn edge detection circuit.....	283
<b>7.6 Basic Function of Timer Array Unit.....</b>	<b>284</b>
7.6.1 Overview of single-operation function and combination-operation function .....	284
7.6.2 Basic rules of combination-operation function.....	284
7.6.3 Applicable range of basic rules of combination-operation function .....	285
<b>7.7 Operation of Timer Array Unit as Independent Channel.....</b>	<b>286</b>
7.7.1 Operation as interval timer/square wave output.....	286
7.7.2 Operation as external event counter .....	294
7.7.3 Operation as frequency divider (channel 0 of unit 0 only) .....	297
7.7.4 Operation as input pulse interval measurement.....	301
7.7.5 Operation as input signal high-/low-level width measurement .....	305
<b>7.8 Operation of Plural Channels of Timer Array Unit.....</b>	<b>309</b>
7.8.1 Operation as PWM function .....	309
7.8.2 Operation as one-shot pulse output function.....	316
7.8.3 Operation as multiple PWM output function .....	323
<b>CHAPTER 8 REAL-TIME COUNTER .....</b>	<b>331</b>
<b>8.1 Functions of Real-Time Counter.....</b>	<b>331</b>
<b>8.2 Configuration of Real-Time Counter .....</b>	<b>331</b>
<b>8.3 Registers Controlling Real-Time Counter .....</b>	<b>333</b>
<b>8.4 Real-Time Counter Operation .....</b>	<b>348</b>
8.4.1 Starting operation of real-time counter.....	348
8.4.2 Shifting to STOP mode after starting operation.....	349
8.4.3 Reading/writing real-time counter .....	350
8.4.4 Setting alarm of real-time counter .....	352
8.4.5 1 Hz output of real-time counter.....	353
8.4.6 32.768 kHz output of real-time counter .....	353
8.4.7 512 Hz, 16.384 kHz output of real-time counter.....	353
8.4.8 Example of watch error correction of real-time counter.....	354

<b>CHAPTER 9 WATCHDOG TIMER .....</b>	<b>359</b>
<b>9.1 Functions of Watchdog Timer.....</b>	<b>359</b>
<b>9.2 Configuration of Watchdog Timer .....</b>	<b>360</b>
<b>9.3 Register Controlling Watchdog Timer.....</b>	<b>361</b>
<b>9.4 Operation of Watchdog Timer.....</b>	<b>362</b>
9.4.1 Controlling operation of watchdog timer .....	362
9.4.2 Setting overflow time of watchdog timer.....	363
9.4.3 Setting window open period of watchdog timer .....	364
9.4.4 Setting watchdog timer interval interrupt .....	365
 <b>CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER.....</b>	 <b>366</b>
<b>10.1 Functions of Clock Output/Buzzer Output Controller .....</b>	<b>366</b>
<b>10.2 Configuration of Clock Output/Buzzer Output Controller.....</b>	<b>367</b>
<b>10.3 Registers Controlling Clock Output/Buzzer Output Controller .....</b>	<b>367</b>
<b>10.4 Operations of Clock Output/Buzzer Output Controller .....</b>	<b>369</b>
10.4.1 Operation as output pin .....	369
 <b>CHAPTER 11 A/D CONVERTER .....</b>	 <b>370</b>
<b>11.1 Function of A/D Converter.....</b>	<b>370</b>
<b>11.2 Configuration of A/D Converter .....</b>	<b>371</b>
<b>11.3 Registers Used in A/D Converter.....</b>	<b>373</b>
<b>11.4 A/D Converter Operations .....</b>	<b>382</b>
11.4.1 Basic operations of A/D converter.....	382
11.4.2 Input voltage and conversion results .....	384
11.4.3 A/D converter operation mode .....	385
<b>11.5 Temperature Sensor Function (Expanded-Specification Products (<math>\mu</math>PD78F117xA) Only).....</b>	<b>387</b>
11.5.1 Configuration of temperature sensor.....	387
11.5.2 Registers used by temperature sensors.....	388
11.5.3 Temperature sensor operation .....	390
11.5.4 Procedures for using temperature sensors .....	392
<b>11.6 How to Read A/D Converter Characteristics Table.....</b>	<b>395</b>
<b>11.7 Cautions for A/D Converter .....</b>	<b>397</b>
 <b>CHAPTER 12 D/A CONVERTER .....</b>	 <b>402</b>
<b>12.1 Function of D/A Converter.....</b>	<b>402</b>
<b>12.2 Configuration of D/A Converter .....</b>	<b>402</b>
<b>12.3 Registers Used in D/A Converter.....</b>	<b>404</b>
<b>12.4 Operation of D/A Converter.....</b>	<b>407</b>
12.4.1 Operation in normal mode.....	407
12.4.2 Operation in real-time output mode .....	408
12.4.3 Cautions .....	409

<b>CHAPTER 13 SERIAL ARRAY UNIT .....</b>	<b>410</b>
<b>13.1 Functions of Serial Array Unit .....</b>	<b>410</b>
13.1.1 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) .....	410
13.1.2 UART (UART0, UART1, UART2, UART3) .....	411
13.1.3 Simplified I <sup>2</sup> C (IIC10, IIC11, IIC20, IIC21) .....	412
<b>13.2 Configuration of Serial Array Unit.....</b>	<b>413</b>
<b>13.3 Registers Controlling Serial Array Unit .....</b>	<b>418</b>
<b>13.4 Operation stop mode .....</b>	<b>441</b>
13.4.1 Stopping the operation by units.....	441
13.4.2 Stopping the operation by channels.....	442
<b>13.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) Communication ....</b>	<b>444</b>
13.5.1 Master transmission .....	445
13.5.2 Master reception .....	454
13.5.3 Master transmission/reception .....	462
13.5.4 Slave transmission .....	470
13.5.5 Slave reception .....	479
13.5.6 Slave transmission/reception .....	485
13.5.7 Calculating transfer clock frequency .....	494
13.5.8 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) communication .....	496
<b>13.6 Operation of UART (UART0, UART1, UART2, UART3) Communication.....</b>	<b>497</b>
13.6.1 UART transmission .....	498
13.6.2 UART reception .....	508
13.6.3 LIN transmission .....	515
13.6.4 LIN reception.....	518
13.6.5 Calculating baud rate .....	523
13.6.6 Procedure for processing errors that occurred during UART (UART0, UART1, UART2, UART3) communication .....	527
<b>13.7 Operation of Simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21) Communication.....</b>	<b>528</b>
13.7.1 Address field transmission .....	529
13.7.2 Data transmission .....	534
13.7.3 Data reception.....	537
13.7.4 Stop condition generation .....	541
13.7.5 Calculating transfer rate.....	542
13.7.6 Procedure for processing errors that occurred during simplified I <sup>2</sup> C (IIC10, IIC11, IIC20, IIC21) communication .....	545
<b>13.8 Relationship Between Register Settings and Pins .....</b>	<b>546</b>

<b>CHAPTER 14 SERIAL INTERFACE IIC0</b> .....	<b>555</b>
<b>14.1 Functions of Serial Interface IIC0</b> .....	<b>555</b>
<b>14.2 Configuration of Serial Interface IIC0</b> .....	<b>558</b>
<b>14.3 Registers to Controlling Serial Interface IIC0</b> .....	<b>561</b>
<b>14.4 I<sup>2</sup>C Bus Mode Functions</b> .....	<b>573</b>
14.4.1 Pin configuration .....	573
<b>14.5 I<sup>2</sup>C Bus Definitions and Control Methods</b> .....	<b>574</b>
14.5.1 Start conditions .....	574
14.5.2 Addresses .....	575
14.5.3 Transfer direction specification.....	575
14.5.4 Transfer clock setting method .....	576
14.5.5 Acknowledge (ACK) .....	577
14.5.6 Stop condition .....	579
14.5.7 Wait .....	580
14.5.8 Canceling wait .....	582
14.5.9 Interrupt request (INTIIC0) generation timing and wait control.....	583
14.5.10 Address match detection method .....	584
14.5.11 Error detection.....	584
14.5.12 Extension code.....	584
14.5.13 Arbitration.....	585
14.5.14 Wakeup function .....	586
14.5.15 Communication reservation.....	587
14.5.16 Cautions .....	591
14.5.17 Communication operations.....	592
14.5.18 Timing of I <sup>2</sup> C interrupt request (INTIIC0) occurrence.....	600
<b>14.6 Timing Charts</b> .....	<b>621</b>
 <b>CHAPTER 15 MULTIPLIER</b> .....	 <b>628</b>
<b>15.1 Functions of Multiplier</b> .....	<b>628</b>
<b>15.2 Configuration of Multiplier</b> .....	<b>629</b>
<b>15.3 Operation of Multiplier</b> .....	<b>630</b>
 <b>CHAPTER 16 DMA CONTROLLER</b> .....	 <b>631</b>
<b>16.1 Functions of DMA Controller</b> .....	<b>631</b>
<b>16.2 Configuration of DMA Controller</b> .....	<b>632</b>
<b>16.3 Registers to Controlling DMA Controller</b> .....	<b>635</b>
<b>16.4 Operation of DMA Controller</b> .....	<b>638</b>
16.4.1 Operation procedure .....	638
16.4.2 Transfer mode.....	639
16.4.3 Termination of DMA transfer .....	639
<b>16.5 Example of Setting of DMA Controller</b> .....	<b>640</b>
16.5.1 CSI consecutive transmission .....	640
16.5.2 CSI master reception.....	642
16.5.3 CSI transmission/reception .....	644
16.5.4 Consecutive capturing of A/D conversion results .....	646
16.5.5 UART consecutive reception + ACK transmission .....	648



16.5.6 Holding DMA transfer pending by DWAITn.....	650
16.5.7 Forced termination by software.....	651
<b>16.6 Cautions on Using DMA Controller.....</b>	<b>652</b>
<b>CHAPTER 17 INTERRUPT FUNCTIONS.....</b>	<b>655</b>
<b>17.1 Interrupt Function Types.....</b>	<b>655</b>
<b>17.2 Interrupt Sources and Configuration.....</b>	<b>655</b>
<b>17.3 Registers Controlling Interrupt Functions.....</b>	<b>661</b>
<b>17.4 Interrupt Servicing Operations.....</b>	<b>671</b>
17.4.1 Maskable interrupt acknowledgment.....	671
17.4.2 Software interrupt request acknowledgment.....	673
17.4.3 Multiple interrupt servicing.....	674
17.4.4 Interrupt request hold.....	677
<b>CHAPTER 18 KEY INTERRUPT FUNCTION.....</b>	<b>678</b>
<b>18.1 Functions of Key Interrupt.....</b>	<b>678</b>
<b>18.2 Configuration of Key Interrupt.....</b>	<b>678</b>
<b>18.3 Register Controlling Key Interrupt.....</b>	<b>679</b>
<b>CHAPTER 19 STANDBY FUNCTION.....</b>	<b>680</b>
<b>19.1 Standby Function and Configuration.....</b>	<b>680</b>
19.1.1 Standby function.....	680
19.1.2 Registers controlling standby function.....	680
<b>19.2 Standby Function Operation.....</b>	<b>683</b>
19.2.1 HALT mode.....	683
19.2.2 STOP mode.....	688
<b>CHAPTER 20 RESET FUNCTION.....</b>	<b>695</b>
<b>20.1 Register for Confirming Reset Source.....</b>	<b>703</b>
<b>CHAPTER 21 POWER-ON-CLEAR CIRCUIT.....</b>	<b>704</b>
<b>21.1 Functions of Power-on-Clear Circuit.....</b>	<b>704</b>
<b>21.2 Configuration of Power-on-Clear Circuit.....</b>	<b>705</b>
<b>21.3 Operation of Power-on-Clear Circuit.....</b>	<b>705</b>
<b>21.4 Cautions for Power-on-Clear Circuit.....</b>	<b>708</b>
<b>CHAPTER 22 LOW-VOLTAGE DETECTOR.....</b>	<b>710</b>
<b>22.1 Functions of Low-Voltage Detector.....</b>	<b>710</b>
<b>22.2 Configuration of Low-Voltage Detector.....</b>	<b>711</b>
<b>22.3 Registers Controlling Low-Voltage Detector.....</b>	<b>711</b>
<b>22.4 Operation of Low-Voltage Detector.....</b>	<b>716</b>
22.4.1 When used as reset.....	717
22.4.2 When used as interrupt.....	723
<b>22.5 Cautions for Low-Voltage Detector.....</b>	<b>729</b>

<b>CHAPTER 23 REGULATOR .....</b>	<b>733</b>
<b>23.1 Regulator Overview.....</b>	<b>733</b>
<b>23.2 Registers Controlling Regulator.....</b>	<b>733</b>
<b>CHAPTER 24 OPTION BYTE.....</b>	<b>735</b>
<b>24.1 Functions of Option Bytes .....</b>	<b>735</b>
24.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H) .....	735
24.1.2 On-chip debug option byte (000C3H/ 010C3H).....	736
<b>24.2 Format of User Option Byte .....</b>	<b>736</b>
<b>24.3 Format of On-chip Debug Option Byte.....</b>	<b>738</b>
<b>24.4 Setting of Option Byte .....</b>	<b>739</b>
<b>CHAPTER 25 FLASH MEMORY .....</b>	<b>740</b>
<b>25.1 Writing with Flash Memory Programmer .....</b>	<b>740</b>
<b>25.2 Programming Environment.....</b>	<b>743</b>
<b>25.3 Communication Mode.....</b>	<b>743</b>
<b>25.4 Connection of Pins on Board.....</b>	<b>744</b>
25.4.1 FLMD0 pin.....	744
25.4.2 TOOL0 pin.....	745
25.4.3 RESET pin .....	745
25.4.4 Port pins .....	746
25.4.5 REGC pin .....	746
25.4.6 X1 and X2 pins .....	746
25.4.7 Power supply.....	746
<b>25.5 Registers that Control Flash Memory .....</b>	<b>746</b>
<b>25.6 Programming Method .....</b>	<b>747</b>
25.6.1 Controlling flash memory.....	747
25.6.2 Flash memory programming mode .....	747
25.6.3 Selecting communication mode .....	748
25.6.4 Communication commands.....	748
<b>25.7 Security Settings .....</b>	<b>750</b>
<b>25.8 Processing Time of Each Command When Using PG-FP4 or PG-FP5 (Reference Values)....</b>	<b>752</b>
<b>25.9 Flash Memory Programming by Self-Programming .....</b>	<b>753</b>
25.9.1 Boot swap function .....	755
25.9.2 Flash shield window function.....	757
<b>CHAPTER 26 ON-CHIP DEBUG FUNCTION .....</b>	<b>758</b>
<b>26.1 Connecting QB-MINI2 to 78K0R/KH3.....</b>	<b>758</b>
<b>26.2 On-Chip Debug Security ID .....</b>	<b>759</b>
<b>26.3 Securing of user resources.....</b>	<b>759</b>
<b>CHAPTER 27 BCD CORRECTION CIRCUIT .....</b>	<b>761</b>
<b>27.1 BCD Correction Circuit Function.....</b>	<b>761</b>
<b>27.2 Registers Used by BCD Correction Circuit .....</b>	<b>761</b>
<b>27.3 BCD Correction Circuit Operation .....</b>	<b>762</b>

<b>CHAPTER 28 INSTRUCTION SET .....</b>	<b>764</b>
<b>28.1 Conventions Used in Operation List.....</b>	<b>765</b>
28.1.1 Operand identifiers and specification methods .....	765
28.1.2 Description of operation column.....	766
28.1.3 Description of flag operation column.....	767
28.1.4 PREFIX Instruction .....	767
<b>28.2 Operation List.....</b>	<b>768</b>
<b>CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS).....</b>	<b>785</b>
<b>CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS).....</b>	<b>845</b>
<b>CHAPTER 31 PACKAGE DRAWINGS.....</b>	<b>903</b>
<b>CHAPTER 32 RECOMMENDED SOLDERING CONDITIONS.....</b>	<b>904</b>
<b>APPENDIX A DEVELOPMENT TOOLS .....</b>	<b>905</b>
<b>A.1 Software Package.....</b>	<b>908</b>
<b>A.2 Language Processing Software.....</b>	<b>908</b>
<b>A.3 Control Software.....</b>	<b>909</b>
<b>A.4 Flash Memory Programming Tools .....</b>	<b>910</b>
A.4.1 When using flash memory programmers PG-FP5, FL-PR5, PG-FP4 and FL-PR4.....	910
A.4.2 When using on-chip debug emulator with programming function QB-MINI2 .....	910
<b>A.5 Debugging Tools (Hardware) .....</b>	<b>911</b>
A.5.1 When using in-circuit emulator QB-78K0RKX3 .....	911
A.5.2 When using on-chip debug emulator with programming function QB-MINI2 .....	912
<b>A.6 Debugging Tools (Software).....</b>	<b>912</b>
<b>APPENDIX B LIST OF CAUTIONS.....</b>	<b>913</b>
<b>APPENDIX C REVISION HISTORY .....</b>	<b>948</b>
<b>C.1 Major Revisions in This Edition .....</b>	<b>948</b>
<b>C.2 Revision History of Preceding Editions .....</b>	<b>953</b>

## CHAPTER 1 OUTLINE

### 1.1 Differences Between Conventional-Specification Products ( $\mu$ PD78F117x) and Expanded-Specification Products ( $\mu$ PD78F117xA)

This manual describes the functions of the 78K0R/KH3 microcontroller products with conventional specifications ( $\mu$ PD78F117x) and expanded specifications ( $\mu$ PD78F117xA).

The differences between the conventional-specification products ( $\mu$ PD78F117x) and expanded-specification products ( $\mu$ PD78F117xA) of the 78K0R/KH3 microcontrollers are described below.

(1/2)

Item	Conditions	Conventional-Specification Products	Expanded-Specification Products	Reference in This Manual
Temperature sensor function	Channel 0 and channel 1 of the A/D converter are used. Internal high-speed oscillator operating	None	Available	11.5 Temperature Sensor Function
Expansion of frequency range of conversion clock ( $f_{AD}$ ) in A/D converter (support of low-speed conversion time)	When the LV1 and LV0 bits of the A/D converter mode register (ADM) are set to 0	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ $f_{AD} = 0.6\text{ to }3.6\text{ MHz}$ $2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$ $f_{AD} = 0.6\text{ to }1.8\text{ MHz}$	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ $f_{AD} = 0.33\text{ to }3.6\text{ MHz}$ $2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$ $f_{AD} = 0.33\text{ to }1.8\text{ MHz}$	11.3 (2) A/D converter mode register (ADM)
Improvement of A/D converter conversion accuracy	Overall error when $2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$ Zero-scale error, full-scale error, integral linearity error, and differential linearity error when $2.3\text{ V} \leq AV_{REF0} < 4.0\text{ V}$	-	Improved	CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS), A/D Converter Characteristics
<R> Number of rewrites	Used for updating programs When using flash memory programmer and NEC Electronics self programming library	100	1000	CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS), Flash Memory Programming Characteristics
Expansion of EEPROM <sup>TM</sup> emulation data retention period	Used for updating data. When EEPROM emulation library provided by NEC Electronics is used (usable ROM size: 6 KB, which consists of 3 consecutive blocks)	3 years	5 years	
Expansion of operating voltage in simplified I <sup>2</sup> C mode (serial array unit)	$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$ , during communication at same potential	Not supported	Supported	CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS), Serial Interface, (d) During communication at same potential (simplified I <sup>2</sup> C mode)

(2/2)

Item	Conditions	Conventional-Specification Products	Expanded-Specification Products	Reference in This Manual
Expansion of operating voltage of external bus interface	$1.8\text{ V} \leq V_{\text{DD}} < 2.7\text{ V}$ , synchronous separate/synchronous multiplexed/asynchronous separate mode	Not supported	Supported	CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS), External Bus Interface
Support for (A) grade product specifications	–	Not supported	Supported	CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)

## 1.2 Features

- Minimum instruction execution time can be changed from high speed (0.05  $\mu$ s: @ 20 MHz operation with high-speed system clock) to ultra low-speed (61  $\mu$ s: @ 32.768 kHz operation with subsystem clock)
- General-purpose register: 8 bits  $\times$  32 registers (8 bits  $\times$  8 registers  $\times$  4 banks)
- ROM, RAM capacities

Part Number	Item	Program Memory (ROM)		Data Memory (RAM)
$\mu$ PD78F1174	Flash memory	128 KB	8 KB	
$\mu$ PD78F1174A				
$\mu$ PD78F1175		192 KB	10 KB	
$\mu$ PD78F1175A				
$\mu$ PD78F1176		256 KB	12 KB	
$\mu$ PD78F1176A				
$\mu$ PD78F1177		384 KB	24 KB	
$\mu$ PD78F1177A				
$\mu$ PD78F1178		512 KB	30 KB	
$\mu$ PD78F1178A				

- On-chip single-power-supply flash memory (with prohibition of chip erase/block erase/writing function)
- Self-programming (with boot swap function/flash shield window function)
- On-chip debug function
- On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)
- On-chip watchdog timer (operable with the on-chip internal low-speed oscillation clock)
- On-chip multiplier (16 bits  $\times$  16 bits)
- On-chip functions of external bus interface
- On-chip key interrupt function
- On-chip clock output/buzzer output controller
- On-chip BCD adjustment
- I/O ports: 116 (N-ch open drain: 4)
- Timer: 14 channels
  - 16-bit timer: 12 channels
  - Watchdog timer: 1 channel
  - Real-time counter: 1 channel
- Serial interface
  - CSI: 2 channels/UART: 1 channel
  - CSI: 2 channels/UART: 1 channel/simplified I<sup>2</sup>C: 2 channels
  - CSI: 2 channels/UART: 1 channel/simplified I<sup>2</sup>C: 2 channels
  - UART (LIN-bus supported): 1 channel
  - I<sup>2</sup>C: 1 channel
- 10-bit resolution A/D converter ( $AV_{REF0} = 2.3$  to 5.5 V): 16 channels
- 8-bit resolution D/A converter ( $AV_{REF1} = 1.8$  to 5.5 V): 2 channels
- Power supply voltage:  $V_{DD} = 1.8$  to 5.5 V
- Operating ambient temperature:  $T_A = -40$  to +85°C

### 1.3 Applications

- Home appliances
  - Laser printer motors
  - Clothes washers
  - Air conditioners
  - Refrigerators
- Home audio systems
- Digital cameras, digital video cameras

### 1.4 Ordering Information

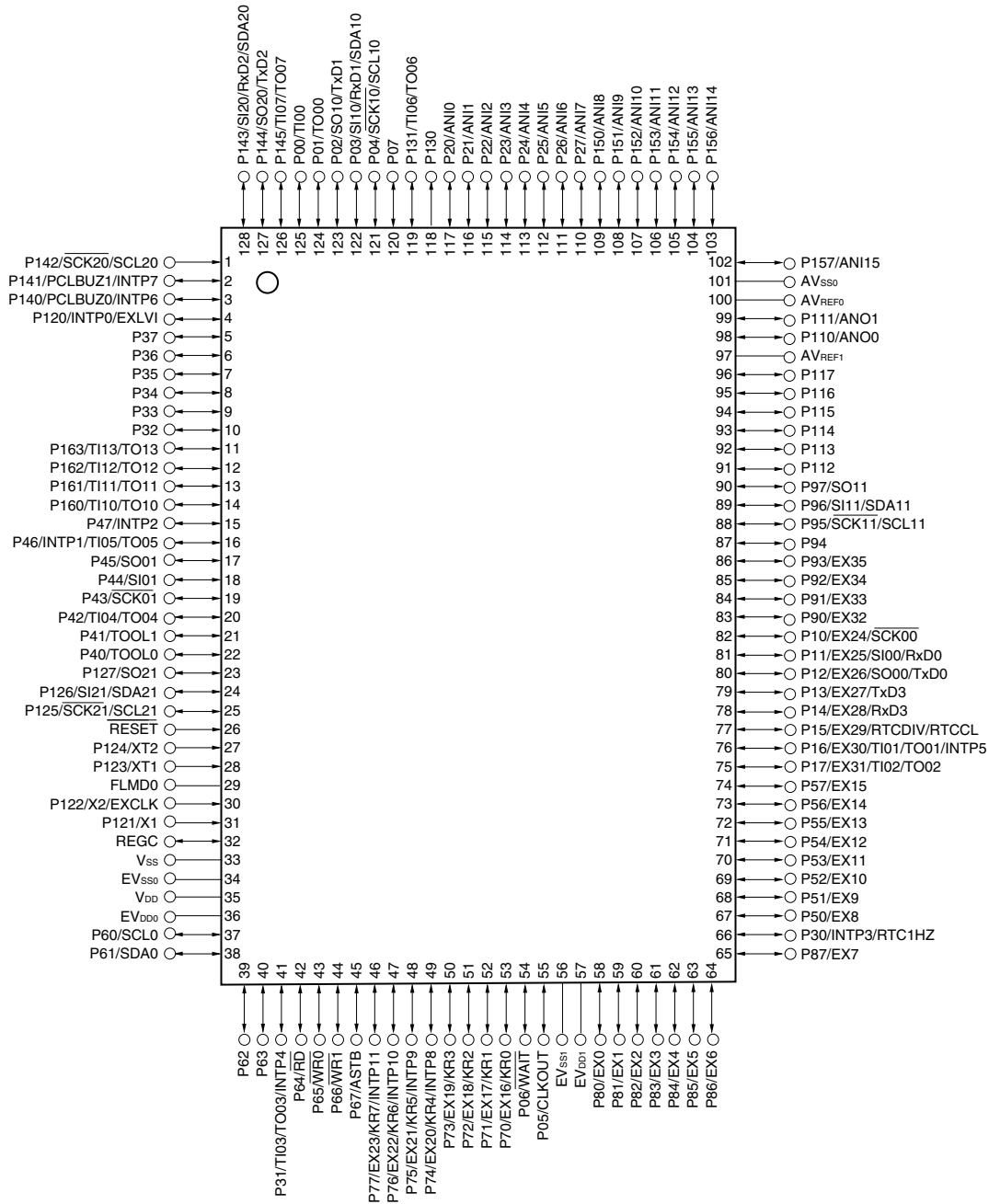
- **Flash memory version**

Part Number	Package	Quality Grade
μPD78F1174GF-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Standard
μPD78F1174AGF-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Standard
μPD78F1175GF-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Standard
μPD78F1175AGF-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Standard
μPD78F1176GF-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Standard
μPD78F1176AGF-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Standard
μPD78F1177GF-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Standard
μPD78F1177AGF-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Standard
μPD78F1178GF-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Standard
μPD78F1178AGF-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Standard
μPD78F1174AGF(A)-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Special
μPD78F1175AGF(A)-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Special
μPD78F1176AGF(A)-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Special
μPD78F1177AGF(A)-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Special
μPD78F1178AGF(A)-GAT-AX	128-pin plastic LQFP (fine pitch) (14 × 20)	Special

Please refer to "Quality Grades on NEC Semiconductor Devices" (Document No. C11531E) published by NEC Electronics Corporation to know the specification of the quality grade on the device and its recommended applications.

### 1.5 Pin Configuration (Top View)

- 128-pin plastic LQFP (fine pitch) (14 × 20)



- Cautions**
1. Make AV<sub>SS</sub>, EV<sub>SS0</sub>, and EV<sub>SS1</sub> the same potential as V<sub>SS</sub>.
  2. Make EV<sub>DD0</sub> and EV<sub>DD1</sub> the same potential as V<sub>DD</sub>.
  3. Connect the REGC pin to V<sub>SS</sub> via a capacitor (0.47 to 1 μF).
  4. P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 are set as analog inputs in the order of P157/ANI15, ..., P150/ANI8, P27/ANI7, ..., P20/ANI0 by the A/D port configuration register (ADPC). When using P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 as analog inputs, start designing from P157/ANI15 (see 11.3 (6) A/D port configuration register (ADPC) for details).

**Remark** When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to supply separate powers to the V<sub>DD</sub> and two EV<sub>DD</sub> pins and connect the V<sub>SS</sub> and two EV<sub>SS</sub> pins to separate ground lines.



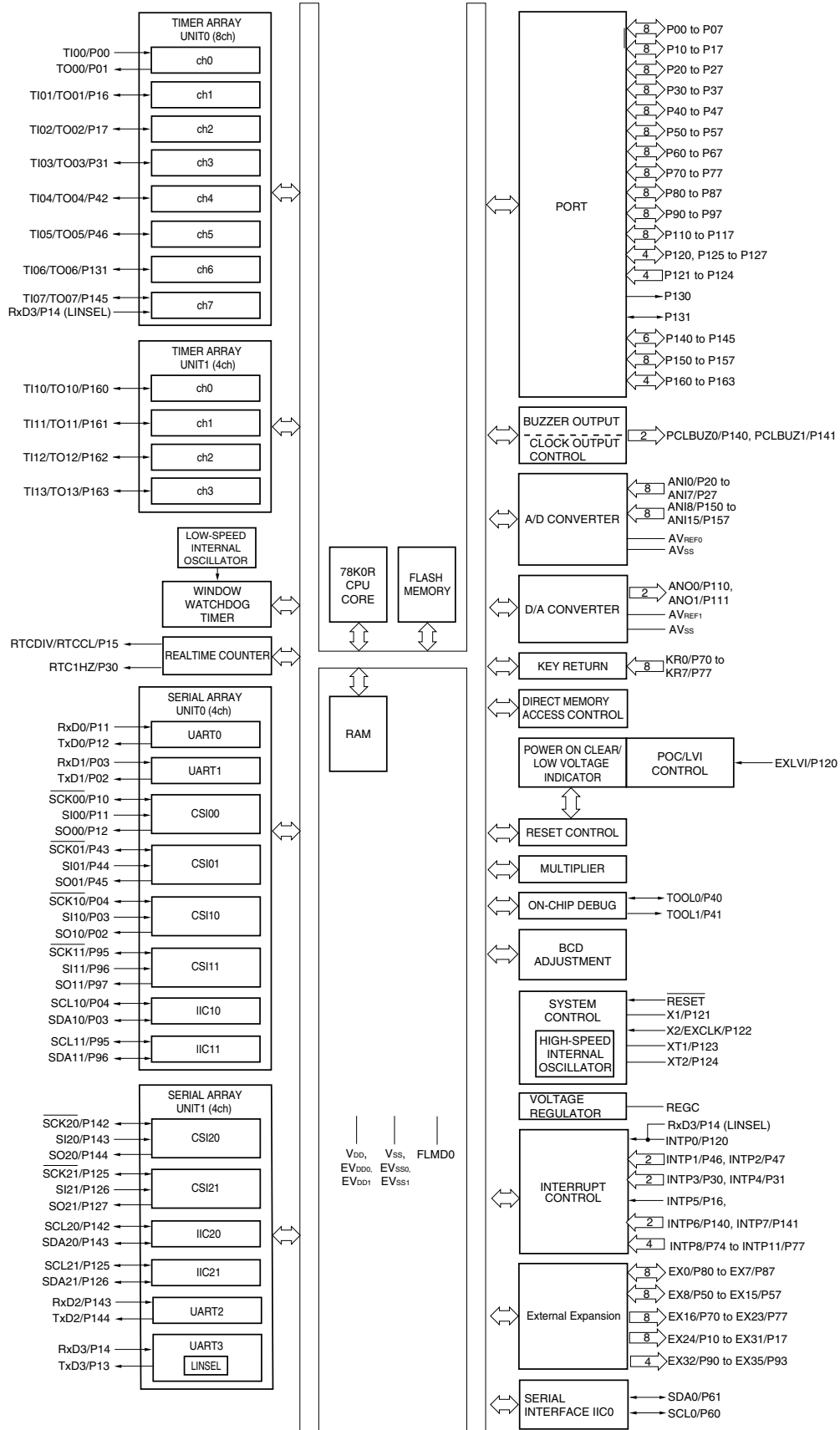
**Pin Identification**

ANI0 to ANI15:	Analog input	$\overline{\text{RESET}}$ :	Reset
ANO0, ANO1:	Analog output	RTC1HZ:	Real-time counter correction clock (1 Hz) output
ASTB:	Address strobe	RTCCL:	Real-time counter clock (32 kHz original oscillation) output
AVREF0, AVREF1:	Analog reference voltage	RTCDIV:	Real-time counter clock (32 kHz divided frequency) output
AVSS :	Analog ground	RxD0 to RxD3:	Receive data
CLKOUT:	Clock output	$\overline{\text{SCK00}}$ , $\overline{\text{SCK01}}$ , $\overline{\text{SCK10}}$ , $\overline{\text{SCK11}}$ , $\overline{\text{SCK20}}$ , $\overline{\text{SCK21}}$ :	Serial clock input/output
EVDD0, EVDD1:	Power supply for port	SCL0, SCL10, SCL11, SCL20, SCL21:	Serial clock input/output
EVSS0, EVSS1:	Ground for port	SDA0, SDA10, SDA11, SDA20, SDA21:	Serial data input/output
EX0 to EX35:	External extension bus	SI00, SI01, SI10, SI11, SI20, SI21:	Serial data input
EXCLK:	External clock input (main system clock)	SO00, SO01, SO10, SO11, SO20, SO21:	Serial data output
EXLVI:	External potential input for low-voltage detector	TI00 to TI07, TI10 to TI13:	Timer input
FLMD0:	Flash programming mode	TO00 to TO07, TO10 to TO13:	Timer output
INTP0 to INTP11:	External interrupt input	TOOL0:	Data input/output for tool
KR0 to KR7:	Key return	TOOL1:	Clock output for tool
P00 to P07:	Port 0	TxD0 to TxD3:	Transmit data
P10 to P17:	Port 1	VDD:	Power supply
P20 to P27:	Port 2	VSS:	Ground
P30 to P37:	Port 3	$\overline{\text{WAIT}}$ :	Wait
P40 to P47:	Port 4	$\overline{\text{WR0}}$ :	Lower byte write strobe
P50 to P57:	Port 5	$\overline{\text{WR1}}$ :	Upper byte write strobe
P60 to P67:	Port 6	X1, X2:	Crystal oscillator (main system clock)
P70 to P77:	Port 7	XT1, XT2:	Crystal oscillator (subsystem clock)
P80 to P87:	Port 8		
P90 to P97:	Port 9		
P110 to P117:	Port 11		
P120 to P127:	Port 12		
P130, P131:	Port 13		
P140 to P145:	Port 14		
P150 to P157:	Port 15		
P160 to P163:	Port 16		
PCLBUZ0, PCLBUZ1:	Programmable clock output/ buzzer output		
$\overline{\text{RD}}$ :	Read strobe		
REGC:	Regulator capacitance		

1.6 78K0R/Kx3 Microcontroller Lineup

ROM	RAM	78K0R/KE3	78K0R/KF3	78K0R/KG3	78K0R/KH3	78K0R/KJ3
		64 Pins	80 Pins	100 Pins	128 Pins	144 Pins
512 KB	30 KB	-	-	$\mu$ PD78F1168	$\mu$ PD78F1178	$\mu$ PD78F1188A
				$\mu$ PD78F1168A	$\mu$ PD78F1178A	
384 KB	24 KB	-	-	$\mu$ PD78F1167	$\mu$ PD78F1177	$\mu$ PD78F1187A
				$\mu$ PD78F1167A	$\mu$ PD78F1177A	
256 KB	12 KB	$\mu$ PD78F1146	$\mu$ PD78F1156	$\mu$ PD78F1166	$\mu$ PD78F1176	$\mu$ PD78F1186A
		$\mu$ PD78F1146A	$\mu$ PD78F1156A	$\mu$ PD78F1166A	$\mu$ PD78F1176A	
192 KB	10 KB	$\mu$ PD78F1145	$\mu$ PD78F1155	$\mu$ PD78F1165	$\mu$ PD78F1175	$\mu$ PD78F1185A
		$\mu$ PD78F1145A	$\mu$ PD78F1155A	$\mu$ PD78F1165A	$\mu$ PD78F1175A	
128 KB	8 KB	$\mu$ PD78F1144	$\mu$ PD78F1154	$\mu$ PD78F1164	$\mu$ PD78F1174	$\mu$ PD78F1184A
		$\mu$ PD78F1144A	$\mu$ PD78F1154A	$\mu$ PD78F1164A	$\mu$ PD78F1174A	
96 KB	6 KB	$\mu$ PD78F1143	$\mu$ PD78F1153	$\mu$ PD78F1163	-	-
		$\mu$ PD78F1143A	$\mu$ PD78F1153A	$\mu$ PD78F1163A		
64 KB	4 KB	$\mu$ PD78F1142	$\mu$ PD78F1152	$\mu$ PD78F1162	-	-
		$\mu$ PD78F1142A	$\mu$ PD78F1152A	$\mu$ PD78F1162A		

1.7 Block Diagram



1.8 Outline of Functions

(1/2)

Item		$\mu$ PD78F1174, $\mu$ PD78F1174A	$\mu$ PD78F1175, $\mu$ PD78F1175A	$\mu$ PD78F1176, $\mu$ PD78F1176A	$\mu$ PD78F1177, $\mu$ PD78F1177A	$\mu$ PD78F1178, $\mu$ PD78F1178A
Internal memory	Flash memory (self-programming supported)	128 KB	192 KB	256 KB	384 KB	512 KB
	RAM	8 KB	10 KB	12 KB	24 KB	30 KB
Memory space		1 MB				
External memory expansion space		824 KB max.	760 KB max.	696 KB max.	568 KB max.	440 KB max.
Main system clock (Oscillation frequency)	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 2 to 20 MHz: $V_{DD} = 2.7$ to 5.5 V, 2 to 5 MHz: $V_{DD} = 1.8$ to 5.5 V				
	Internal high-speed oscillation clock	Internal oscillation 8 MHz (TYP.): $V_{DD} = 1.8$ to 5.5 V				
Subsystem clock (Oscillation frequency)		XT1 (crystal) oscillation 32.768 kHz (TYP.): $V_{DD} = 1.8$ to 5.5 V				
Internal low-speed oscillation clock (For WDT)		Internal oscillation 240 kHz (TYP.): $V_{DD} = 1.8$ to 5.5 V				
General-purpose register		8 bits $\times$ 32 registers (8 bits $\times$ 8 registers $\times$ 4 banks)				
Minimum instruction execution time		0.05 $\mu$ s (High-speed system clock: $f_{MX} = 20$ MHz operation)				
		0.125 $\mu$ s (Internal high-speed oscillation clock: $f_{IH} = 8$ MHz (TYP.) operation)				
		61 $\mu$ s (Subsystem clock: $f_{SUB} = 32.768$ kHz operation)				
Instruction set		<ul style="list-style-type: none"> <li>8-bit operation, 16-bit operation</li> <li>Multiply (8 bits <math>\times</math> 8 bits)</li> <li>Bit manipulation (Set, reset, test, and Boolean operation), etc.</li> </ul>				
I/O port		Total: 116 CMOS I/O: 107 CMOS input: 4 CMOS output: 1 N-ch open-drain I/O (6 V tolerance): 4				
Timer		<ul style="list-style-type: none"> <li>16-bit timer: 12 channels</li> <li>Watchdog timer: 1 channel</li> <li>Real-time counter: 1 channel</li> </ul>				
	Timer outputs	12 (PWM outputs: timer array unit 0: 7, timer array unit 1: 3)				
	RTC outputs	2 <ul style="list-style-type: none"> <li>1 Hz (Subsystem clock: <math>f_{SUB} = 32.768</math> kHz)</li> <li>512 Hz or 16.384 kHz or 32.768 kHz (Subsystem clock: <math>f_{SUB} = 32.768</math> kHz)</li> </ul>				
Clock output/buzzer output		2 <ul style="list-style-type: none"> <li>2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (peripheral hardware clock: <math>f_{MAIN} = 20</math> MHz operation)</li> <li>256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (Subsystem clock: <math>f_{SUB} = 32.768</math> kHz operation)</li> </ul>				
A/D converter		10-bit resolution $\times$ 16 channels ( $AV_{REF0} = 2.3$ to 5.5 V)				
D/A converter		8-bit resolution $\times$ 2 channels ( $AV_{REF1} = 1.8$ to 5.5 V)				

Item	$\mu$ PD78F1174, $\mu$ PD78F1174A	$\mu$ PD78F1175, $\mu$ PD78F1175A	$\mu$ PD78F1176, $\mu$ PD78F1176A	$\mu$ PD78F1177, $\mu$ PD78F1177A	$\mu$ PD78F1178, $\mu$ PD78F1178A
Serial interface	<ul style="list-style-type: none"> <li>• UART supporting LIN-bus: 1 channel</li> <li>• CSI: 2 channels/UART: 1 channel</li> <li>• CSI: 2 channels/UART: 1 channel/simplified I<sup>2</sup>C: 2 channels</li> <li>• CSI: 2 channels/UART: 1 channel/simplified I<sup>2</sup>C: 2 channels</li> <li>• I<sup>2</sup>C bus: 1 channel</li> </ul>				
Multiplier	16 bits $\times$ 16 bits = 32 bits				
DMA controller	2 channels				
Vectored interrupt sources	Internal	32			
	External	13			
Key interrupt	Key interrupt (INTKR) occurs by detecting falling edge of the key input pins (KR0 to KR7).				
Reset	<ul style="list-style-type: none"> <li>• Reset by <math>\overline{\text{RESET}}</math> pin</li> <li>• Internal reset by watchdog timer</li> <li>• Internal reset by power-on-clear</li> <li>• Internal reset by low-voltage detector</li> <li>• Internal reset by illegal instruction execution<sup>Note</sup></li> </ul>				
On-chip debug function	Provided				
Power supply voltage	$V_{DD} = 1.8$ to 5.5 V				
Operating ambient temperature	$T_A = -40$ to $+85^\circ\text{C}$				
Package	128-pin plastic LQFP (fine pitch) (14 $\times$ 20) (0.5 mm pitch)				

**Note** When instruction code FFH is executed.

Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

## CHAPTER 2 PIN FUNCTIONS

### 2.1 Pin Function List

There are five types of pin I/O buffer power supplies:  $AV_{REF0}$ ,  $AV_{REF1}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ , and  $V_{DD}$ . The relationship between these power supplies and the pins is shown below.

**Table 2-1. Pin I/O Buffer Power Supplies**

Power Supply	Corresponding Pins
$AV_{REF0}$	P20 to P27, P150 to P157
$AV_{REF1}$	P110, P111
$EV_{DD0}$ , $EV_{DD1}$	<ul style="list-style-type: none"><li>• Port pins other than P20 to P27, P110, P111, P121 to P124, and P150 to P157</li><li>• <math>\overline{RESET}</math> pin and FLMD0 pin</li></ul>
$V_{DD}$	<ul style="list-style-type: none"><li>• P121 to P124</li><li>• Pins other than port pins (except <math>\overline{RESET}</math> pin and FLMD0 pin )</li></ul>

(1) Port functions (1/3)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 8-bit I/O port. Input of P03 and P04 can be set to TTL input buffer. Output of P02 to P04 can be set to N-ch open-drain output ( $V_{DD}$ tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI00
P01				TO00
P02				SO10/TxD1
P03				SI10/RxD1/SDA10
P04				SCK10/SCL10
P05				CLKOUT
P06				WAIT
P07				–
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCK00/EX24
P11				SI00/RxD0/EX25
P12				SO00/TxD0/EX26
P13				TxD3/EX27
P14				RxD3/EX28
P15				RTCDIV/RTCCL/EX29
P16				TI01/TO01/INTP5/EX30
P17				TI02/TO02/EX31
P20 to P27	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI0 to ANI7
P30	I/O	Port 3. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	RTC1HZ/INTP3
P31				TI03/TO03/INTP4
P32 to P37				–
P40 <sup>Note</sup>	I/O	Port 4. 8-bit I/O port. Input of P43 and P44 can be set to TTL input buffer. Output of P43 and P45 can be set to N-ch open-drain output ( $V_{DD}$ tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOOL0
P41				TOOL1
P42				TI04/TO04
P43				SCK01
P44				SI01
P45				SO01
P46				INTP1/TI05/TO05
P47				INTP2
P50 to P57	I/O	Port 5. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	EX8 to EX15

**Note** If on-chip debugging is enabled by using an option byte, be sure to pull up the P40/TOOL0 pin externally (see **Caution** in 2.2.5 P40 to P47 (port 4)).

(1) Port functions (2/3)

Function Name	I/O	Function	After Reset	Alternate Function
P60	I/O	Port 6. 8-bit I/O port. Output of P60 to P63 can be set to N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units. For only P64 to P67, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCL0
P61				SDA0
P62				–
P63				–
P64				$\overline{\text{RD}}$
P65				$\overline{\text{WR0}}$
P66				$\overline{\text{WR1}}$
P67				ASTB
P70 to P73	I/O	Port 7. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	KR0/EX16 to KR3/EX19
P74 to P77				KR4/EX20/INTP8 to KR7/EX23/INTP11
P80 to P87	I/O	Port 8. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	EX0 to EX7
P90 to P93	I/O	Port 9. 8-bit I/O port. Input of P95 and P96 can be set to TTL input buffer. Output of P95 to P97 can be set to the N-ch open-drain output ( $V_{\text{DD}}$ tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	EX32 to EX35
P94				–
P95				$\overline{\text{SCK11}}/\text{SCL11}$
P96				SI11/SDA11
P97				SO11
P110	I/O	Port 11. 8-bit I/O port. Input/output can be specified in 1-bit units. For only P112 to P117, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	ANO0
P111				ANO1
P112 to P117				–
P120	I/O	Port 12. 4-bit I/O port and 4-bit input port. Input of P125 and P126 can be set to TTL input buffer. Output of P125 to P127 can be set to the N-ch open-drain output ( $V_{\text{DD}}$ tolerance). For only P120 and P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121				X1
P122				X2/EXCLK
P123				XT1
P124	I/O			XT2
P125				$\overline{\text{SCK21}}/\text{SCL21}$
P126				SI21/SDA21
P127				SO21



(1) Port functions (3/3)

Function Name	I/O	Function	After Reset	Alternate Function
P130	Output	Port 13.	Output port	–
P131	I/O	1-bit output port and 1-bit I/O port. For only P131 use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI06/TO06
P140	I/O	Port 14. 6-bit I/O port. Input of P142 and P143 can be set to TTL input buffer. Output of P142 to P144 can be set to the N-ch open-drain output ( $V_{DD}$ tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	PCLBUZ0/INTP6
P141				PCLBUZ1/INTP7
P142				SCK20/SCL20
P143				SI20/RxD2/SDA20
P144				SO20/TxD2
P145				TI07/TO07
P150 to P157	I/O	Port 15. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI8 to ANI15
P160 to P163	I/O	Port 16. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI10/TO10 to TI13/TO13

(2) Non-port functions (1/4)

Function Name	I/O	Function	After Reset	Alternate Function			
ANI0 to ANI7	Input	A/D converter analog input	Digital input port	P20 to P27			
ANI8 to ANI15	Input	A/D converter analog input	Digital input port	P150 to P157			
ANO0	Output	D/A converter analog output	Input port	P110			
ANO1	Output	D/A converter analog output	Input port	P111			
CLKOUT	Output	External expansion clock output	Input port	P05			
$\overline{\text{WAIT}}$	Input	External wait input	Input port	P06			
$\overline{\text{RD}}$	Output	Read strobe signal output to external memory	Input port	P64			
$\overline{\text{WR0}}$	Output	Write strobe to external memory (lower 8-bit)	Input port	P65			
$\overline{\text{WR1}}$	Output	Write strobe to external memory (higher 8-bit)	Input port	P66			
ASTB	Output	Address strobe signal output to external memory	Input port	P67			
EX0 to EX7	I/O	External expansion I/O	Input port	P80 to P87			
EX8 to EX15				P50 to P57			
EX16 to EX19	Output	External expansion output	Input port	P70/KR0 to P73/KR3			
EX20 to EX23				P74/KR4/INTP8 to P77/KR7/INTP11			
EX24				P10/SCK00			
EX25				P11/RxD0/SI00			
EX26				P12/TxD0/SO00			
EX27				P13/TxD3			
EX28				P14/RxD3			
EX29				P15/RTCDIV/RTCCL			
EX30				P16/TI01/TO01/INTP5			
EX31				P17/TI02/TO02			
EX32 to EX35				P90 to P93			
EXLVI				Input	Potential input for external low-voltage detection	Input port	P120/INTP0
INTP0				Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P120/EXLVI
INTP1	P46/TI05/TO05						
INTP2	P47						
INTP3	P30/RTC1HZ						
INTP4	P31/TI03/TO03						
INTP5	P16/TI01/TO01/EX30						
INTP6	P140/PCLBUZ0						
INTP7	P141/PCLBUZ1						
INTP8	P74/KR4/EX20 to P77/KR7/EX23						
INTP9							
INTP10							
INTP11							
KR0 to KR3	Input	Key interrupt input	Input port	P70/EX16 to P73/EX19			
KR4 to KR7				P74/EX20/INTP8 to P77/EX23/INTP11			

(2) Non-port functions (2/4)

Function Name	I/O	Function	After Reset	Alternate Function
PCLBUZ0	Output	Clock output/buzzer output	Input port	P140/INTP6
PCLBUZ1				P141/INTP7
REGC	–	Connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect to V <sub>ss</sub> via a capacitor (0.47 to 1 $\mu$ F).	–	–
RTCDIV	Output	Real-time counter clock (32 kHz divided frequency) output	Input port	P15/RTCCL/EX29
RTCCL	Output	Real-time counter clock (32 kHz original oscillation) output	Input port	P15/RTCDIV/EX29
RTC1HZ	Output	Real-time counter correction clock (1 Hz) output	Input port	P30/INTP3
RESET	Input	System reset input	–	–
RxD0	Input	Serial data input to UART0	Input port	P11/SI00/EX25
RxD1		Serial data input to UART1		P03/SI10/SDA10
RxD2		Serial data input to UART2		P143/SI20/SDA20
RxD3		Serial data input to UART3		P14/EX28
SCK00	I/O	Clock input/output for CSI00, CSI01, CSI10, CSI11, CSI20, and CSI21	Input port	P10/EX24
SCK01				P43
SCK10				P04/SCL10
SCK11				P95/SCL11
SCK20				P142/SCL20
SCK21				P125/SCL21
SCL0	I/O	Clock input/output for I <sup>2</sup> C	Input port	P60
SCL10	I/O	Clock input/output for simplified I <sup>2</sup> C	Input port	P04/SCK10
SCL11				P95/SCK11
SCL20				P142/SCK20
SCL21				P125/SCK21
SDA0	I/O	Serial data I/O for I <sup>2</sup> C	Input port	P61
SDA10	I/O	Serial data I/O for simplified I <sup>2</sup> C	Input port	P03/SI10/RxD1
SDA11				P96/SI11
SDA20				P143/SI20/RxD2
SDA21				P126/SI21
SI00	Input	Serial data input to CSI00, CSI01, CSI10, CSI11, CSI20, and CSI21	Input port	P11/RxD0/EX25
SI01				P44
SI10				P03/RxD1/SDA10
SI11				P96/SDA11
SI20				P143/RxD2/SDA20
SI21				P126/SDA21
SO00	Output	Serial data output from CSI00, CSI01, CSI10, CSI11, CSI20, and CSI21	Input port	P12/TxD0/EX26
SO01				P45
SO10				P02/TxD1
SO11				P97
SO20				P144/TxD2
SO21				P127

(2) Non-port functions (3/4)

Function Name	I/O	Function	After Reset	Alternate Function
Ti00	Input	External count clock input to 16-bit timer 00	Input port	P00
Ti01		External count clock input to 16-bit timer 01		P16/TO01/INTP5/EX30
Ti02		External count clock input to 16-bit timer 02		P17/TO02/EX31
Ti03		External count clock input to 16-bit timer 03		P31/TO03/INTP4
Ti04		External count clock input to 16-bit timer 04		P42/TO04
Ti05		External count clock input to 16-bit timer 05		P46/INTP1/TO05
Ti06		External count clock input to 16-bit timer 06		P131/TO06
Ti07		External count clock input to 16-bit timer 07		P145/TO07
Ti10		External count clock input to 16-bit timer 10		P160/TO10
Ti11		External count clock input to 16-bit timer 11		P161/TO11
Ti12		External count clock input to 16-bit timer 12		P162/TO12
Ti13		External count clock input to 16-bit timer 13		P163/TO13
TO00		Output		16-bit timer 00 output
TO01	16-bit timer 01 output		P16/Ti01/INTP5/EX30	
TO02	16-bit timer 02 output		P17/Ti02/EX31	
TO03	16-bit timer 03 output		P31/Ti03/INTP4	
TO04	16-bit timer 04 output		P42/Ti04	
TO05	16-bit timer 05 output		P46/INTP1/Ti05	
TO06	16-bit timer 06 output		P131/Ti06	
TO07	16-bit timer 07 output		P145/Ti07	
TO10	16-bit timer 10 output		P160/Ti10	
TO11	16-bit timer 11 output		P161/Ti11	
TO12	16-bit timer 12 output		P162/Ti12	
TO13	16-bit timer 13 output		P163/Ti13	
TxD0	Output		Serial data output from UART0	Input port
TxD1		Serial data output from UART1	P02/SO10	
TxD2		Serial data output from UART2	P144/SO20	
TxD3		Serial data output from UART3	P13/EX27	
X1	–	Resonator connection for main system clock	Input port	P121
X2	–		Input port	P122/EXCLK
EXCLK	Input	External clock input for main system clock	Input port	P122/X2
XT1	–	Resonator connection for subsystem clock	Input port	P123
XT2	–		Input port	P124
V <sub>DD</sub>	–	Positive power supply (P121 to P124 and other than ports (excluding $\overline{\text{RESET}}$ and FLMD0 pins))	–	–
EV <sub>DD0</sub> , EV <sub>DD1</sub>	–	Positive power supply for ports (other than P20 to P27, P110, P111, P121 to P124, P150 to P157) and $\overline{\text{RESET}}$ and FLMD0 pins	–	–
AV <sub>REF0</sub>	–	<ul style="list-style-type: none"> <li>A/D converter reference voltage input</li> <li>Positive power supply for P20 to P27, P150 to P157, and A/D converter</li> </ul>	–	–
AV <sub>REF1</sub>	–	<ul style="list-style-type: none"> <li>D/A converter reference voltage input</li> <li>Positive power supply for P110, P111, and D/A converter</li> </ul>	–	–

**(2) Non-port functions (4/4)**

Function Name	I/O	Function	After Reset	Alternate Function
V <sub>SS</sub>	–	Ground potential (P121 to P124 and other than ports (excluding $\overline{\text{RESET}}$ and FLMD0 pins))	–	–
EV <sub>SS0</sub> , EV <sub>SS1</sub>	–	Ground potential for ports (other than P20 to P27, P110, P111, P121 to P124, and P150 to P157) and $\overline{\text{RESET}}$ and FLMD0 pins	–	–
AV <sub>SS</sub>	–	Ground potential for A/D converter, D/A converter, P20 to P27, P110, P111, and P150 to P157. Make this pin the same potential as the EV <sub>SS0</sub> , EV <sub>SS1</sub> , or V <sub>SS</sub> .	–	–
FLMD0	–	Flash memory programming mode setting	–	–
TOOL0	I/O	Data I/O for flash memory programmer/debugger	Input port	P40
TOOL1	Output	Clock output for debugger	Input port	P41

## 2.2 Description of Pin Functions

### 2.2.1 P00 to P07 (port 0)

P00 to P07 function as an 8-bit I/O port. P00 to P06 pins also function as timer I/O, serial interface data I/O, clock I/O, internal system clock output, and external wait signal input.

Input to the P03 and P04 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units, using port input mode register 0 (PIM0).

Output from the P02 to P04 pins can be specified as normal CMOS output or N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units, using port output mode register 0 (POM0).

The following operation modes can be specified in 1-bit units.

#### (1) Port mode

P00 to P07 function as an 8-bit I/O port. P00 to P07 can be set to input or output port in 1-bit units using port mode register 0 (PM0). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

#### (2) Control mode

P00 to P06 function as timer I/O, serial interface data I/O, clock I/O, internal system clock output, and external wait signal input.

##### (a) TI00

This is a pin for inputting an external count clock/capture trigger to 16-bit timer 00.

##### (b) TO00

This is a timer output pin of 16-bit timer 00.

##### (c) SI10

This is a serial data input pin of serial interface CSI10.

##### (d) SO10

This is a serial data output pin of serial interface CSI10.

##### (e) $\overline{\text{SCK10}}$

This is a serial clock I/O pin of serial interface CSI10.

##### (f) TxD1

This is a serial data output pin of serial interface UART1.

##### (g) RxD1

This is a serial data input pin of serial interface UART1.

##### (h) SDA10

This is a serial data I/O pin of serial interface for simplified I<sup>2</sup>C.

##### (i) SCL10

This is a serial clock I/O pin of serial interface for simplified I<sup>2</sup>C.

**(j) CLKOUT**

This is an internal system clock output pin.

**(k)  $\overline{\text{WAIT}}$** 

This is an external wait signal input pin.

**Caution** To use P02/SO10/TxD1 and P04/ $\overline{\text{SCK10}}$ /SCL10 as general-purpose ports, set serial communication operation setting register 02 (SCR02) to the default status (0087H). In addition, clear port output mode register 0 (POM0) to 00H.

**2.2.2 P10 to P17 (port 1)**

P10 to P17 function as an 8-bit I/O port. These pins also function as external interrupt request input, serial interface data I/O, clock I/O, timer I/O, real-time counter clock output, and external expansion output.

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P10 to P17 function as an 8-bit I/O port. P10 to P17 can be set to input or output port in 1-bit units using port mode register 1 (PM1). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

**(2) Control mode**

P10 to P17 function as external interrupt request input, serial interface data I/O, clock I/O, timer I/O, real-time counter clock output, and external expansion output.

**(a) SI00**

This is a serial data input pin of serial interface CSI00.

**(b) SO00**

This is a serial data output pin of serial interface CSI00.

**(c)  $\overline{\text{SCK00}}$** 

This is a serial clock I/O pin of serial interface CSI00.

**(d) RxD0**

This is a serial data input pin of serial interface UART0.

**(e) RxD3**

This is a serial data input pin of serial interface UART3.

**(f) TxD0**

This is a serial data output pin of serial interface UART0.

**(g) TxD3**

This is a serial data output pin of serial interface UART3.

**(h) TI01, TI02**

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 01 and 02.

**(i) TO01, TO02**

These are the timer output pins of 16-bit timers 01 and 02.

**(j) INTP5**

This is an external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**(k) RTCDIV**

This is a real-time counter clock (32 kHz, divided) output pin.

**(l) RTCCL**

This is a real-time counter clock (32 kHz, original oscillation) output pin.

**(m) EX24 to EX31**

These are the external expansion output (address bus) pins.

- Cautions**
1. To use P10/SCK00/EX24 and P12/SO00/TxD0/EX26 as general-purpose ports, set serial communication operation setting register 00 (SCR00) to the default status (0087H).
  2. Do not enable outputting RTCCL and RTCDIV at the same time.

**2.2.3 P20 to P27 (port 2)**

P20 to P27 function as an 8-bit I/O port. These pins also function as A/D converter analog input.

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P20 to P27 function as an 8-bit I/O port. P20 to P27 can be set to input or output port in 1-bit units using port mode register 2 (PM2).

**(2) Control mode**

P20 to P27 function as A/D converter analog input pins (ANI0 to ANI7). When using these pins as analog input pins, see 11.7 (6) ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157.

**Caution** ANI0/P20 to ANI7/P27 are set in the digital input (general-purpose port) mode after release of reset.

**2.2.4 P30 to P37 (port 3)**

P30 to P37 function as an 8-bit I/O port. P30 and P31 pins also function as external interrupt request input, timer I/O, and real-time counter correction clock output.

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P30 to P37 function as an 8-bit I/O port. P30 to P37 can be set to input or output port in 1-bit units using port mode register 3 (PM3). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).



**(2) Control mode**

P30 and P31 function as external interrupt request input, timer I/O, and real-time counter correction clock output.

**(a) INTP3, INTP4**

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**(b) TI03**

This is a pin for inputting an external count clock/capture trigger to 16-bit timer 03.

**(c) TO03**

This is a timer output pin from 16-bit timer 03.

**(d) RTC1HZ**

This is a real-time counter correction clock (1 Hz) output pin.

**2.2.5 P40 to P47 (port 4)**

P40 to P47 function as an 8-bit I/O port. These pins also function as external interrupt request input, serial interface data I/O, clock I/O, data I/O for a flash memory programmer/debugger, clock output, and timer I/O.

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

Output from the P43 and P45 pins can be specified as normal CMOS output or N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 4 (POM4).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P40 to P47 function as an 8-bit I/O port. P40 to P47 can be set to input or output port in 1-bit units using port mode register 4 (PM4). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 4 (PU4).

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

**(2) Control mode**

P40 to P47 function as serial interface data I/O, clock I/O, external interrupt request input, data I/O for a flash memory programmer/debugger, clock output, and timer I/O.

**(a) INTP1, INTP2**

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**(b) TOOL0**

This is a data I/O pin for a flash memory programmer/debugger.

Be sure to pull up this pin externally when on-chip debugging is enabled (pulling it down is prohibited).

**(c) TOOL1**

This is a clock output pin for a debugger.

When the on-chip debug function is used, P41/TOOL1 pin can be used as follows by the mode setting on the debugger.

1-line mode: can be used as a port (P41).

2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).

**(d) TI04, TI05**

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 04 and 05.

**(e) TO04, TO05**

These are the timer output pins from 16-bit timers 04 and 05.

**(f)  $\overline{\text{SCK01}}$** 

This is a serial clock I/O pin of serial interface CSI01.

**(g) SI01**

This is a serial data input pin of serial interface CSI01.

**(h) SO01**

This is a serial data output pin of serial interface CSI01.

**Caution** The function of the P40/TOOL0 pin varies as described in (a) to (c) below.  
In the case of (b) or (c), make the specified connection.

**(a) In normal operation mode and when on-chip debugging is disabled (OCDENSET = 0) by an option byte (000C3H)**

=> Use this pin as a port pin (P40).

**(b) In normal operation mode and when on-chip debugging is enabled (OCDENSET = 1) by an option byte (000C3H)**

=> Connect this pin to EV<sub>DD0</sub> or EV<sub>DD1</sub> via an external resistor, and always input a high level to the pin before reset release.

**(c) When on-chip debug function is used, or in write mode of flash memory programmer**

=> Use this pin as TOOL0.

Directly connect this pin to the on-chip debug emulator or a flash memory programmer, or pull it up by connecting it to EV<sub>DD0</sub> or EV<sub>DD1</sub> via an external resistor.

**2.2.6 P50 to P57 (port 5)**

P50 to P57 function as an 8-bit I/O port. These pins also function as external expansion I/O.

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P50 to P57 function as an 8-bit I/O port. P50 to P57 can be set to input or output port in 1-bit units using port mode register 5 (PM5). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 5 (PU5).

**(2) Control mode**

P50 to P57 function as external expansion I/O.

**(a) EX8 to EX15**

These are the external expansion I/O (multiplexed address/data bus, address bus, data bus) pins.

**2.2.7 P60 to P67 (port 6)**

P60 to P67 function as an 8-bit I/O port. P60 and P61, P64 to P67 pins also function as serial interface data I/O, clock I/O, read strobe signal output, write strobe signal output, and address strobe signal output.

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P60 to P67 function as an 8-bit I/O port. P60 to P67 can be set to input port or output port in 1-bit units using port mode register 6 (PM6). Only for P64 to P67, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 6 (PU6).

Output of P60 to P63 is N-ch open-drain output (6 V tolerance).

**(2) Control mode**

P60 and P61, P64 to P67 function as serial interface data I/O, clock I/O, read strobe signal output, write strobe signal output, and address strobe signal output.

**(a) SDA0**

This is a serial data I/O pin of serial interface IIC0.

**(b) SCL0**

This is a serial clock I/O pin of serial interface IIC0.

**(c)  $\overline{RD}$** 

This is a read strobe signal output pin.

**(d)  $\overline{WR0}$** 

This is a write strobe signal output (8-bit bus mode, 16-bit bus mode (lower byte)) pin.

**(e)  $\overline{WR1}$** 

This is a write strobe signal output (16-bit bus mode (higher byte)) pin.

**(f) ASTB**

This is an address strobe signal output pin.

**2.2.8 P70 to P77 (port 7)**

P70 to P77 function as an 8-bit I/O port. These pins also function as key interrupt input, external interrupt request input, and external expansion output.

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P70 to P77 function as an 8-bit I/O port. P70 to P77 can be set to input or output port in 1-bit units using port mode register 7 (PM7). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

**(2) Control mode**

P70 to P77 function as key interrupt input, external interrupt request input, and external expansion output.

**(a) KR0 to KR7**

These are the key interrupt input pins

**(b) INTP8 to INTP11**

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**(c) EX16 to EX23**

These are the external expansion output (address bus) pins.

**2.2.9 P80 to P87 (port 8)**

P80 to P87 function as an 8-bit I/O port. These pins also function as external expansion I/O.

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P80 to P87 function as an 8-bit I/O port. P80 to P87 can be set to input or output port in 1-bit units using port mode register 8 (PM8). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 8 (PU8).

**(2) Control mode**

P80 to P87 function as external expansion I/O.

**(a) EX0 to EX7**

These are the external expansion I/O (multiplexed address/data bus, data bus) pins.

**2.2.10 P90 to P97 (port 9)**

P90 to P97 function as an 8-bit I/O port. P90 to P93 and P95 to P97 pins also function as external expansion output, serial interface data I/O, and clock I/O.

Input to the P95 and P96 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 9 (PIM9).

Output from the P95 to P97 pins can be specified as normal CMOS output or N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 9 (POM9).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P90 to P97 function as an 8-bit I/O port. P90 to P97 can be set to input or output port in 1-bit units using port mode register 9 (PM9). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 9 (PU9).

**(2) Control mode**

P90 to P93 and P95 to P97 function as external expansion output, serial interface data I/O, and clock I/O.

**(a) EX32 to EX35**

These are the external expansion output (address bus) pins.

**(b) SI11**

This is a serial data input pin of serial interface CSI11.

**(c) SO11**

This is a serial data output pin of serial interface CSI11.

**(d)  $\overline{\text{SCK11}}$** 

This is a serial clock I/O pin of serial interface CSI11.

**(e) SDA11**

This is a serial data I/O pin of serial interface for simplified I<sup>2</sup>C.

**(f) SCL11**

This is a serial clock I/O pin of serial interface for simplified I<sup>2</sup>C.

**2.2.11 P110 to P117 (port 11)**

P110 to P117 function as an 8-bit I/O port. P110 and P111 pins also function as D/A converter analog output.

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P110 to P117 function as an 8-bit I/O port. P110 to P117 can be set to input or output port in 1-bit units using port mode register 11 (PM11). Only for P112 to P117, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 11 (PU11).

**(2) Control mode**

P110 and P111 function as D/A converter analog output pins (ANO0, ANO1). When using these pins as analog input pins, see **12.4.3 Cautions**.

**2.2.12 P120 to P127 (port 12)**

P120 and P125 to P127 function as a 4-bit I/O port. P121 to P124 function as a 4-bit input port. These pins also function as external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, serial interface data I/O, and clock I/O.

Input to the P125 and P126 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 12 (PIM12).

Output from the P125 to P127 pins can be specified as normal CMOS output or N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 12 (POM12).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P120 and P125 to P127 function as a 4-bit I/O port. P120 and P125 to P127 can be set to input or output port using port mode register 12 (PM12). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

P121 to P124 function as a 4-bit input port.

**(2) Control mode**

P120 to P127 function as external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, serial interface data I/O, and clock I/O.

**(a) INTP0**

This is an external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**(b) EXLVI**

This is a potential input pin for external low-voltage detection.

**(c) X1, X2**

These are the pins for connecting a resonator for main system clock.

**(d) EXCLK**

This is an external clock input pin for main system clock.

**(e) XT1, XT2**

These are the pins for connecting a resonator for subsystem clock.

**(f) SI21**

This is a serial data input pin of serial interface CSI21.

**(g) SO21**

This is a serial data output pin of serial interface CSI21.

**(h)  $\overline{\text{SCK21}}$** 

This is a serial clock I/O pin of serial interface CSI21.

**(i) SDA21**

This is a serial data I/O pin of serial interface for simplified I<sup>2</sup>C.

**(j) SCL21**

This is a serial clock I/O pin of serial interface for simplified I<sup>2</sup>C.

**2.2.13 P130, P131 (port 13)**

P130 functions as a 1-bit output port. P131 functions as a 1-bit I/O port. P131 pin also function as timer I/O.

**Remark** When the device is reset, P130 outputs a low level. Therefore, to output a high level from P130 before the device is reset, the output signal of P130 can be used as a pseudo reset signal of the CPU (see the figure for **Remark** in **4.2.13 Port 13**).

**(1) Port mode**

P130 functions as a 1-bit output port.

P131 functions as a 1-bit I/O port. P131 can be set to input or output port in 1-bit units using port mode register 13 (PM13). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 13 (PU13).

**(2) Control mode**

P131 functions as timer I/O.

**(a) TI06**

This is a pin for inputting an external count clock/capture trigger to 16-bit timer 06.

**(b) TO06**

This is a timer output pin from 16-bit timer 06.

**2.2.14 P140 to P145 (port 14)**

P140 to P145 function as a 6-bit I/O port. P140 P145 pins also function as timer I/O, external interrupt request input, clock/buzzer output, serial interface data I/O, and clock I/O.

Input to the P142 and P143 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 14 (PIM14).

Output from the P142 to P144 pins can be specified as normal CMOS output or N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 14 (POM14).

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P140 to P145 function as a 6-bit I/O port. P140 to P145 can be set to input or output port in 1-bit units using port mode register 14 (PM14). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 14 (PU14).

**(2) Control mode**

P140 to P145 function as timer I/O, external interrupt request input, clock/buzzer output, serial interface data I/O, and clock I/O.

**(a) INT6, INT7**

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

**(b) PCLBUZ0, PCLBUZ1**

These are the clock/buzzer output pins.

**(c) TI07**

This is a pin for inputting an external count clock/capture trigger to 16-bit timer 07.

**(d) TO07**

This is a timer output pin of 16-bit timer 07.

**(e) SI20**

This is a serial data input pin of serial interface CSI20.

**(f) SO20**

This is a serial data output pin of serial interface CSI20.

**(g)  $\overline{\text{SCK20}}$** 

This is a serial clock I/O pin of serial interface CSI20.

**(h) TxD2**

This is a serial data output pin of serial interface UART2.

**(i) RxD2**

This is a serial data input pin of serial interface UART2.

**(j) SDA20**

This is a serial data I/O pin of serial interface for simplified I<sup>2</sup>C.

**(k) SCL20**

This is a serial clock I/O pin of serial interface for simplified I<sup>2</sup>C.

**2.2.15 P150 to P157 (port 15)**

P150 to P157 function as an 8-bit I/O port. These pins also function as A/D converter analog input.

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P150 to P157 function as an 8-bit I/O port. P150 to P157 can be set to input or output port in 1-bit units using port mode register 15 (PM15).

**(2) Control mode**

P150 to P157 function as A/D converter analog input pins (ANI8 to ANI15). When using these pins as analog input pins, see 11.7 (6) **ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157**.

**Caution** ANI8/P150 to ANI15/P157 are set in the digital input (general-purpose port) mode after release of reset.

**2.2.16 P160 to P163 (port 16)**

P160 to P163 function as a 4-bit I/O port. These pins also function as timer I/O.

The following operation modes can be specified in 1-bit units.

**(1) Port mode**

P160 to P163 function as a 4-bit I/O port. P160 to P163 can be set to input or output port in 1-bit units using port mode register 16 (PM16). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 16 (PU16).



**(2) Control mode**

P160 to P163 function as timer I/O.

**(a) TI10 to TI13**

These are the pins for inputting an external count clock/capture trigger to 16-bit timer 10 to 13.

**(b) TO10 to TO13**

These are the timer output pins of 16-bit timer 10 to 13.

**2.2.17 AV<sub>REF0</sub>**

This is the A/D converter reference voltage input pin and the positive power supply pin of P20 to P27, P150 to P157, and A/D converter.

The voltage that can be supplied to AV<sub>REF0</sub> varies as follows, depending on whether P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 are used as digital I/Os or analog inputs.

**Table 2-2. AV<sub>REF0</sub> Voltage Applied to P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 Pins**

Analog/Digital	V <sub>DD</sub> Condition	AV <sub>REF0</sub> Voltage
Using at least one pin as an analog input and using all pins not as digital I/Os	2.3 V ≤ V <sub>DD</sub> ≤ 5.5 V	2.3 V ≤ AV <sub>REF0</sub> ≤ V <sub>DD</sub> = EV <sub>DD0</sub> = EV <sub>DD1</sub>
Pins used as analog inputs and digital I/Os are mixed <sup>Note</sup>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2.7 V ≤ AV <sub>REF0</sub> ≤ V <sub>DD</sub> = EV <sub>DD0</sub> = EV <sub>DD1</sub>
	2.3 V ≤ V <sub>DD</sub> < 2.7 V	AV <sub>REF0</sub> has same potential as EV <sub>DD0</sub> , EV <sub>DD1</sub> , and V <sub>DD</sub>
Using at least one pin as a digital I/O and using all pins not as analog inputs <sup>Note</sup>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2.7 V ≤ AV <sub>REF0</sub> ≤ V <sub>DD</sub> = EV <sub>DD0</sub> = EV <sub>DD1</sub>
	1.8 V ≤ V <sub>DD</sub> < 2.7 V	AV <sub>REF0</sub> has same potential as EV <sub>DD0</sub> , EV <sub>DD1</sub> , and V <sub>DD</sub>

**Note** AV<sub>REF0</sub> is the reference for the I/O voltage of a port to be used as a digital port.

- High-/low-level input voltage (V<sub>IH4</sub>/V<sub>IL4</sub>)
- High-/low-level output voltage (V<sub>OH2</sub>/V<sub>OL2</sub>)

**2.2.18 AVREF1**

This is the D/A converter reference voltage input pin and the positive power supply pin of P110, P111, and the D/A converter.

The voltage that can be supplied to AVREF1 varies as follows, depending on whether P110/ANO0 and P111/ANO1 are used as digital I/Os or analog outputs.

**Table 2-3. AVREF1 Voltage Applied to P110/ANO0 and P111/ANO1 Pins**

Analog/Digital	V <sub>DD</sub> Condition	AVREF1 Voltage
Using at least one pin as an analog output and using all pins not as digital I/Os	1.8 V ≤ V <sub>DD</sub> ≤ 5.5 V	1.8 V ≤ AVREF1 ≤ V <sub>DD</sub> = EV <sub>DD0</sub> = EV <sub>DD1</sub>
Pins used as analog outputs and digital I/Os are mixed <sup>Note</sup>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2.7 V ≤ AVREF1 ≤ V <sub>DD</sub> = EV <sub>DD0</sub> = EV <sub>DD1</sub>
	1.8 V ≤ V <sub>DD</sub> < 2.7 V	AVREF1 has same potential as EV <sub>DD0</sub> , EV <sub>DD1</sub> , and V <sub>DD</sub>
Using at least one pin as a digital I/O and using all pins not as analog outputs <sup>Note</sup>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2.7 V ≤ AVREF1 ≤ V <sub>DD</sub> = EV <sub>DD0</sub> = EV <sub>DD1</sub>
	1.8 V ≤ V <sub>DD</sub> < 2.7 V	AVREF1 has same potential as EV <sub>DD0</sub> , EV <sub>DD1</sub> , and V <sub>DD</sub>

**Note** AVREF1 is the reference for the I/O voltage of a port to be used as a digital port.

- High-/low-level input voltage (V<sub>IH5</sub>/V<sub>IL5</sub>)
- High-/low-level output voltage (V<sub>OH2</sub>/V<sub>OL2</sub>)

**2.2.19 AVss**

This is the ground potential pin of A/D converter, D/A converter, P20 to P27, P110, P111, and P150 to P157. Even when the A/D converter and D/A converter are not used, always use this pin with the same potential as EV<sub>SS0</sub>, EV<sub>SS1</sub>, and V<sub>SS</sub>.

**2.2.20 RESET**

This is the active-low system reset input pin.

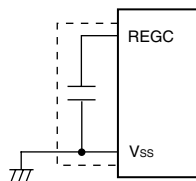
When the external reset pin is not used, connect this pin directly or via a resistor to EV<sub>DD0</sub> or EV<sub>DD1</sub>.

When the external reset pin is used, design the circuit based on V<sub>DD</sub>.

**2.2.21 REGC**

This is the pin for connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect this pin to V<sub>SS</sub> via a capacitor (0.47 to 1 μF). However, when using the STOP mode that has been entered since operation of the internal high-speed oscillation clock and external main system clock, 0.47 μF is recommended.

Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



**Caution** Keep the wiring length as short as possible for the broken-line part in the above figure.

### 2.2.22 V<sub>DD</sub>, EV<sub>DD0</sub>, EV<sub>DD1</sub>

V<sub>DD</sub> is the positive power supply pin for P121 to P124 and pins other than ports (excluding the  $\overline{\text{RESET}}$  and FLMD0 pins).

EV<sub>DD0</sub> and EV<sub>DD1</sub> are the positive power supply pins for ports other than P20 to P27, P110, P111, P121 to P124, and P150 to P157 as well as for the  $\overline{\text{RESET}}$  and FLMD0 pins.

### 2.2.23 V<sub>SS</sub>, EV<sub>SS0</sub>, EV<sub>SS1</sub>

V<sub>SS</sub> is the ground potential pin for P121 to P124 and pins other than ports (excluding the  $\overline{\text{RESET}}$  and FLMD0 pins).

EV<sub>SS0</sub> and EV<sub>SS1</sub> are the ground potential pins for ports other than P20 to P27, P110, P111, P121 to P124, and P150 to P157 as well as for the  $\overline{\text{RESET}}$  and FLMD0 pins.

### 2.2.24 FLMD0

This is a pin for setting flash memory programming mode.

Perform either of the following processing.

#### (a) In normal operation mode

It is recommended to leave this pin open during normal operation.

The FLMD0 pin must always be kept at the V<sub>SS</sub> level before reset release but does not have to be pulled down externally because it is internally pulled down by reset. However, pulling it down must be kept selected (i.e., FLMDPUP = "0", default value) by using bit 7 (FLMDPUP) of the background event control register (BECTL) (see **25.5 (1) Back ground event control register**). To pull it down externally, use a resistor of 200 k $\Omega$  or smaller.

Self programming and the rewriting of flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V<sub>SS</sub> pin.

#### (b) In self programming mode

It is recommended to leave this pin open when using the self programming function. To pull it down externally, use a resistor of 100 k $\Omega$  to 200 k $\Omega$ .

In the self programming mode, the setting is switched to pull up in the self programming library.

#### (c) In flash memory programming mode

Directly connect this pin to a flash memory programmer when data is written by the flash memory programmer. This supplies a writing voltage of the V<sub>DD</sub> level to the FLMD0 pin.

The FLMD0 pin does not have to be pulled down externally because it is internally pulled down by reset. To pull it down externally, use a resistor of 1 k $\Omega$  to 200 k $\Omega$ .

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

Table 2-4 shows the types of pin I/O circuits and the recommended connections of unused pins.

**Table 2-4. Connection of Unused Pins (1/3)**

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI00	8-R	I/O	Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>SS0</sub> , or EV <sub>SS1</sub> via a resistor. Output: Leave open.
P01/TO00	5-AG		
P02/SO10/TxD1			
P03/SI10/RxD1/SDA10	5-AN		
P04/ $\overline{\text{SCK10}}$ /SCL10			
P05/CLKOUT	8-R		
P06/ $\overline{\text{WAIT}}$			
P07	5-AG		
P10/ $\overline{\text{SCK00}}$ /EX24	8-R		
P11/SI00/RxD0/EX25			
P12/SO00/TxD0/EX26	5-AG		
P13/TxD3/EX27			
P14/RxD3/EX28	8-R		
P15/RTCDIV/RTCCL/EX29	5-AG		
P16/TI01/TO01/INTP5/EX30	8-R		
P17/TI02/TO02/EX31			
P20/ANI0 to P27/ANI7 <sup>Note</sup>	11-G		
P30/RTC1HZ/INTP3	8-R	Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>SS0</sub> , or EV <sub>SS1</sub> via a resistor. Output: Leave open.	
P31/TI03/TO03/INTP4			
P32 to P37	5-AG		
P40/TOOL0	8-R	<When on-chip debugging is enabled> Pull this pin up (pulling it down is prohibited). <When on-chip debugging is disabled> Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>SS0</sub> , or EV <sub>SS1</sub> via a resistor. Output: Leave open.	
P41/TOOL1	5-AG	Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>SS0</sub> , or EV <sub>SS1</sub> via a resistor. Output: Leave open.	
P42/TI04/TO04	8-R		
P43/ $\overline{\text{SCK01}}$	5-AN		
P44/SI01			
P45/SO01	5-AG		
P46/TI05/TO05/INTP1	8-R		
P47/INTP2			

**Note** P20/ANI0 to P27/ANI7 are set in the digital input port mode after release of reset.

Table 2-4. Connection of Unused Pins (2/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins	
P50/EX8, P51/EX9	8-R	I/O	Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>SS0</sub> , or EV <sub>SS1</sub> via a resistor. Output: Leave open.	
P52/EX10 to P57/EX15	5-AG			
P60/SCL0	13-R		Input: Connect to EV <sub>SS0</sub> or EV <sub>SS1</sub> . Output: Set the port output latch to 0 and leave these pins open via low-level output.	
P61/SDA0				
P62, P63	13-P		Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>SS0</sub> , or EV <sub>SS1</sub> via a resistor. Output: Leave open.	
P64/ $\overline{RD}$	5-AG			
P65/ $\overline{WR0}$				
P66/ $\overline{WR1}$				
P67/ $\overline{ASTB}$				
P70/KR0/EX16 to P73/KR3/EX19	8-R			
P74/KR4/EX20/INTP8 to P77/KR7/EX23/INTP11	5-AG			
P80/EX0 to P87/EX7				
P90/EX32 to P93/EX35				
P94				
P95/ $\overline{SCK11}$ /SCL11	5-AN		Input: Independently connect to AV <sub>REF1</sub> or AV <sub>SS</sub> via a resistor. Output: Leave open.	
P96/SI11/SDA11				
P97/SO11	5-AG			
P110/ANO0, P111/ANO1	12-G			
P112 to P117	5-AG	Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>SS0</sub> , or EV <sub>SS1</sub> via a resistor. Output: Leave open.		
P120/INTP0/EXLVI	8-R			
P121/X1 <sup>Note</sup>	37-B	Input		Independently connect to V <sub>DD</sub> or V <sub>SS</sub> via a resistor.
P122/X2/EXCLK <sup>Note</sup>				
P123/XT1 <sup>Note</sup>				
P124/XT2 <sup>Note</sup>				
P125/ $\overline{SCK21}$ /SCL21	5-AN	I/O		Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>SS0</sub> , or EV <sub>SS1</sub> via a resistor. Output: Leave open.
P126/SI21/SDA21				
P127/SO21	5-AG	Output		Leave open.
P130	3-C			
P131/TI06/TO06	8-R	I/O		Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>SS0</sub> , or EV <sub>SS1</sub> via a resistor. Output: Leave open.

**Notes** Use recommended connection above in input port mode (see **Figure 6-2 Format of Clock Operation Mode Control Register (CMC)**) when these pins are not used.

Table 2-4. Connection of Unused Pins (3/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P140/PCLBUZ0/INTP6	8-R	I/O	Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>SS0</sub> , or EV <sub>SS1</sub> via a resistor. Output: Leave open.
P141/PCLBUZ1/INTP7			
P142/ $\overline{\text{SCK20}}$ /SCL20	5-AN		
P143/SI20/RxD2/SDA20			
P144/SO20/TxD2	5-AG		
P145/TI07/TO07	8-R		
P150/ANI8 to P157/ANI15 <sup>Note</sup>	11-G	Input: Independently connect to AV <sub>REF0</sub> or AV <sub>SS</sub> via a resistor. Output: Leave open.	
P160/TI10/TO10 to P163/TI13/TO13	8-R	Input: Independently connect to EV <sub>DD0</sub> , EV <sub>DD1</sub> , EV <sub>SS0</sub> , or EV <sub>SS1</sub> via a resistor. Output: Leave open.	
AV <sub>REF0</sub>	–	–	Make this pin the same potential as EV <sub>DD0</sub> , EV <sub>DD1</sub> , or V <sub>DD</sub> . See 2.2.17 AV <sub>REF0</sub> when using P20 to P27 and P150 to P157.
AV <sub>REF1</sub>	–	–	Make this pin the same potential as EV <sub>DD0</sub> , EV <sub>DD1</sub> , or V <sub>DD</sub> . See 2.2.18 AV <sub>REF1</sub> when using P110 and P111.
AV <sub>SS</sub>	–	–	Make this pin the same potential as the EV <sub>SS0</sub> , EV <sub>SS1</sub> , or V <sub>SS</sub> .
FLMD0	2-W	–	Leave open or connect to V <sub>SS</sub> via a resistor of 100 k $\Omega$ or more.
$\overline{\text{RESET}}$	2	Input	Connect directly or via a resistor to EV <sub>DD0</sub> or EV <sub>DD1</sub> .
REGC	–	–	Connect to V <sub>SS</sub> via capacitor (0.47 to 1 $\mu\text{F}$ ).

**Notes** P150/ANI8 to P157/ANI15 are set in the digital input port mode after release of reset.

Figure 2-1. Pin I/O Circuit List (1/2)

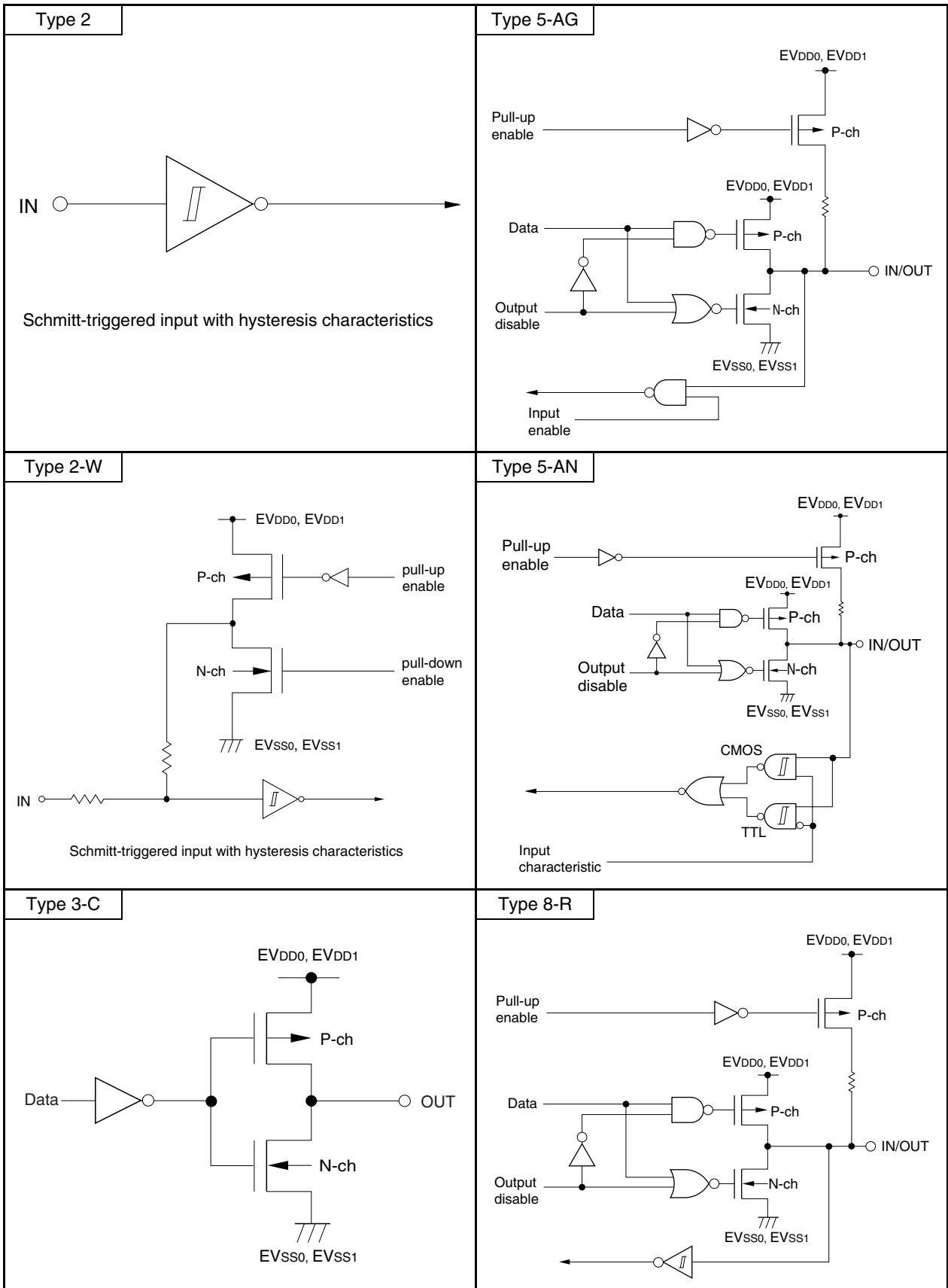
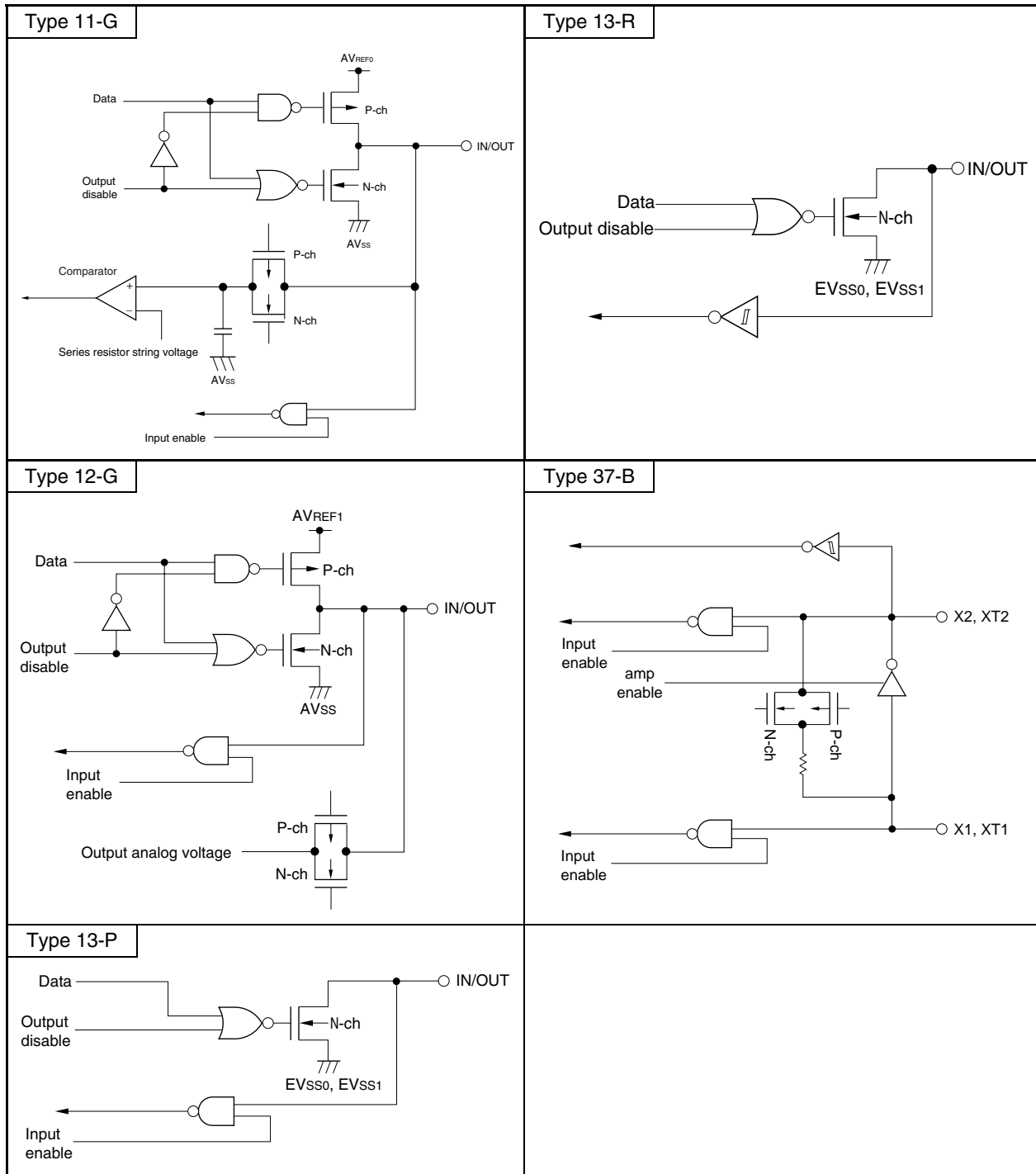


Figure 2-1. Pin I/O Circuit List (2/2)



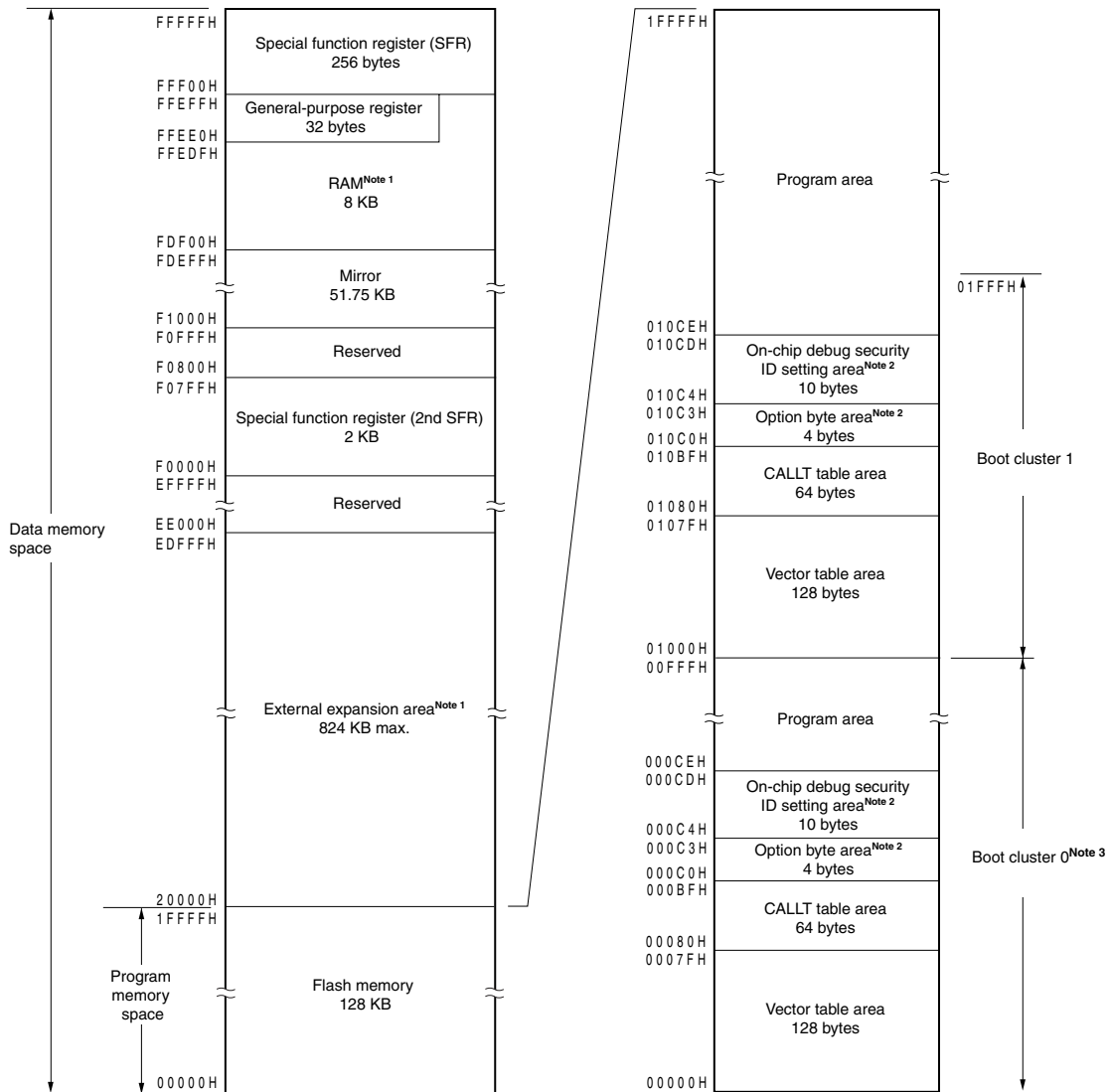


## CHAPTER 3 CPU ARCHITECTURE

### 3.1 Memory Space

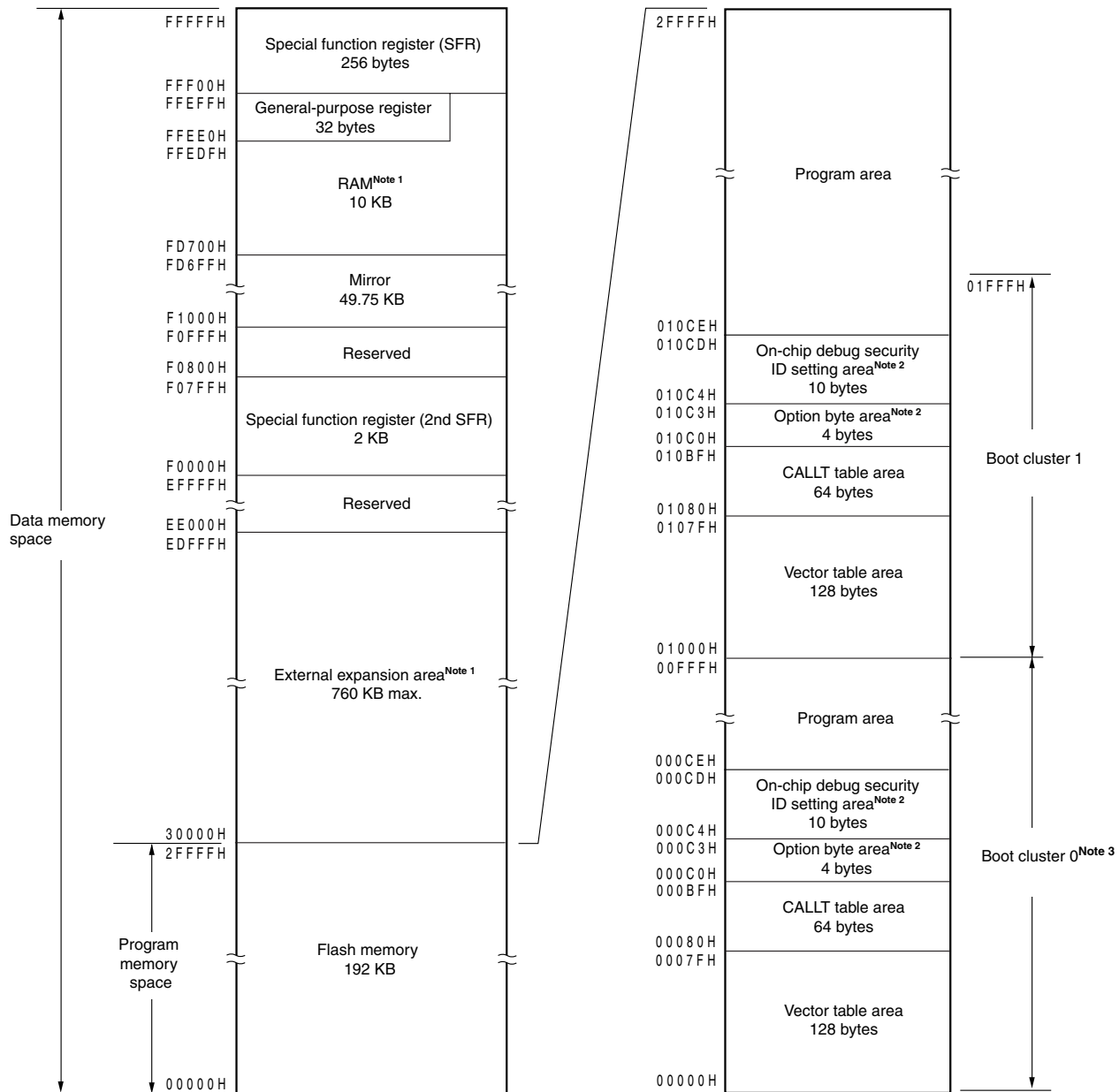
Products in the 78K0R/KH3 can access a 1 MB memory space. Figures 3-1 to 3-5 show the memory maps.

**Figure 3-1. Memory Map ( $\mu$ PD78F1174, 78F1174A)**



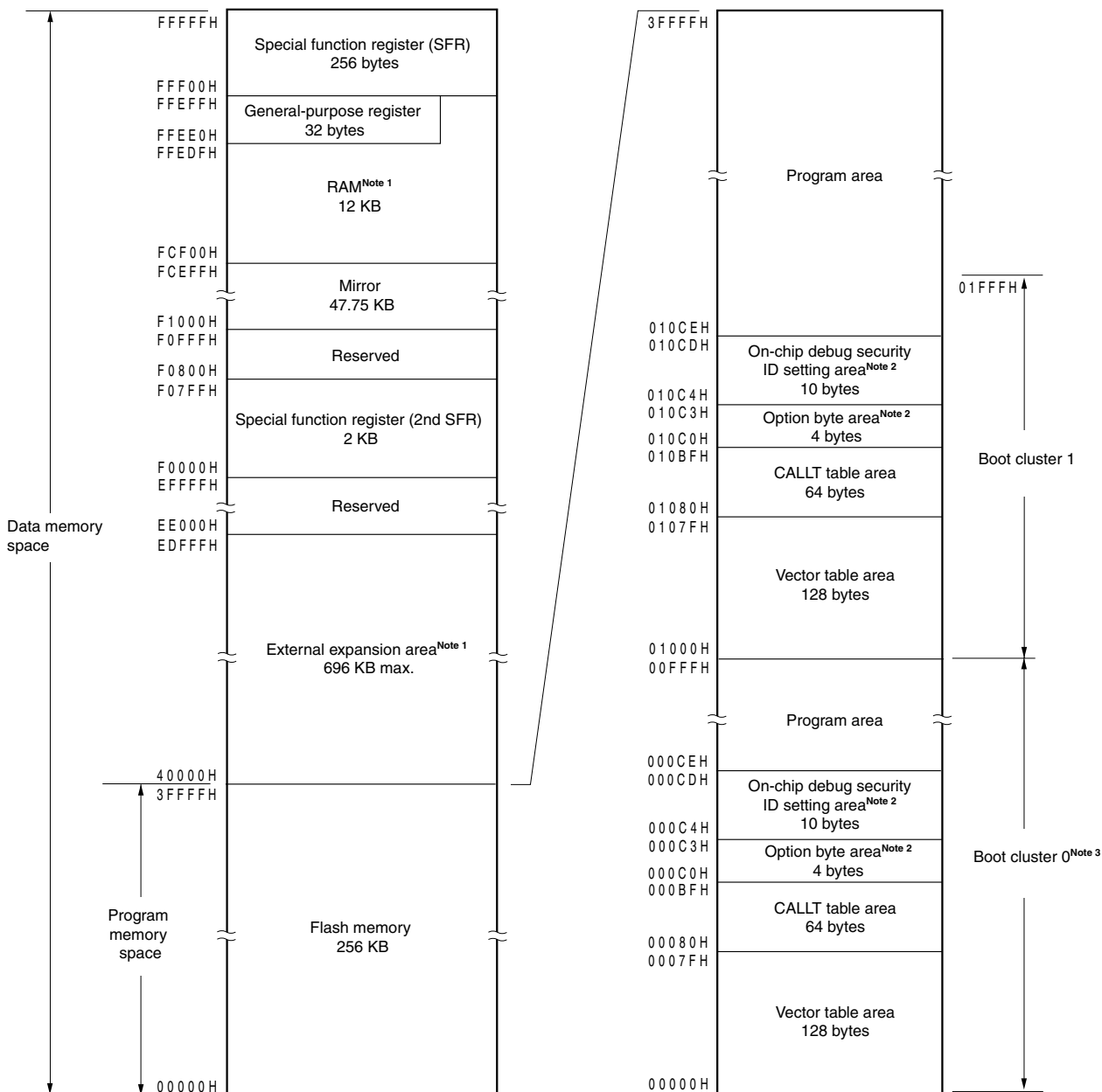
- Notes**
- Instructions can be executed from the RAM area excluding the general-purpose register area, and from the external expansion area.
  - When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.  
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.
  - Writing boot cluster 0 can be prohibited depending on the setting of security (see **25.7 Security Setting**).

Figure 3-2. Memory Map ( $\mu$ PD78F1175, 78F1175A)



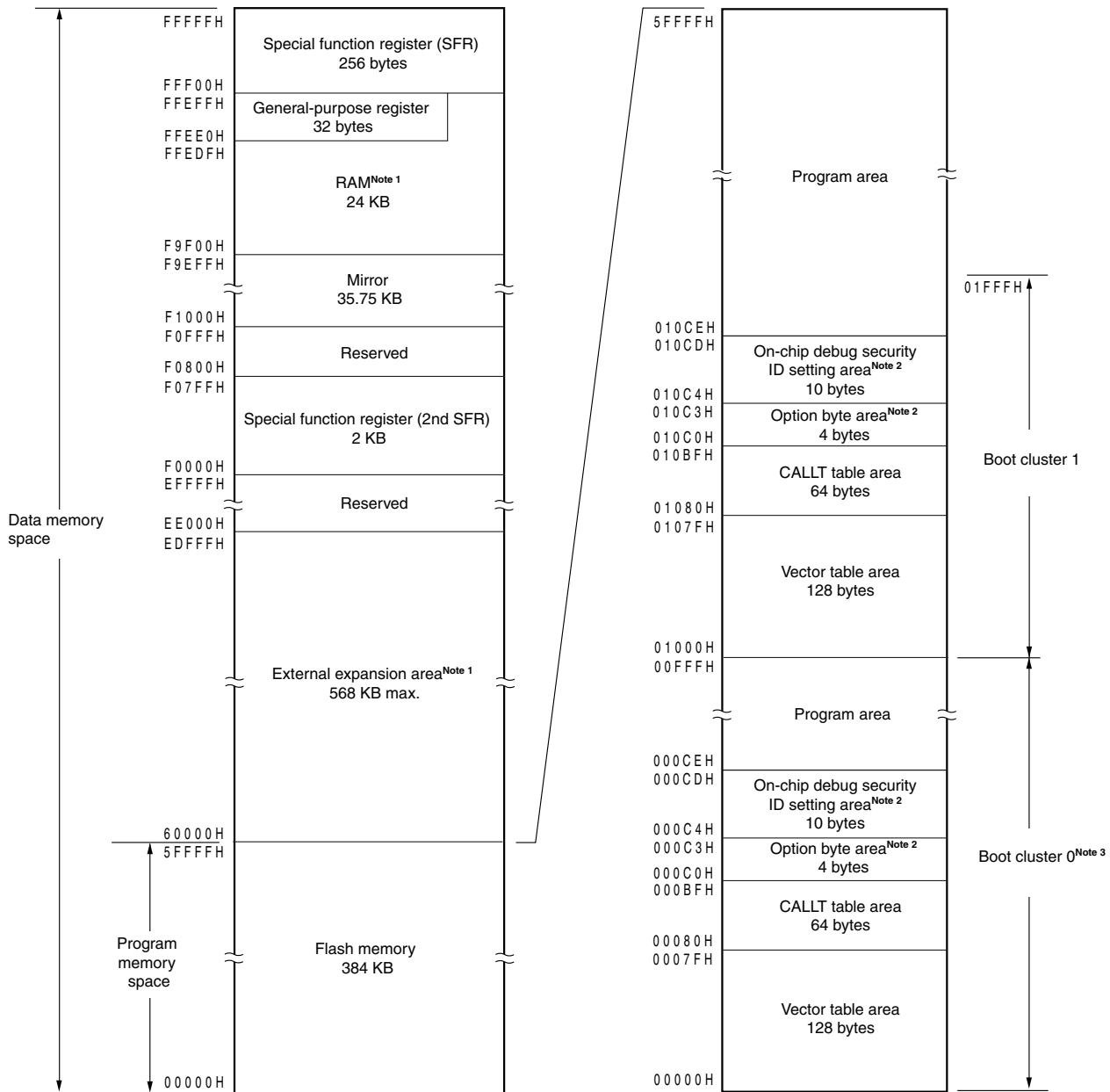
- Notes**
- Instructions can be executed from the RAM area excluding the general-purpose register area, and from the external expansion area.
  - When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.  
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.
  - Writing boot cluster 0 can be prohibited depending on the setting of security (see **25.7 Security Setting**).

Figure 3-3. Memory Map ( $\mu$ PD78F1176, 78F1176A)



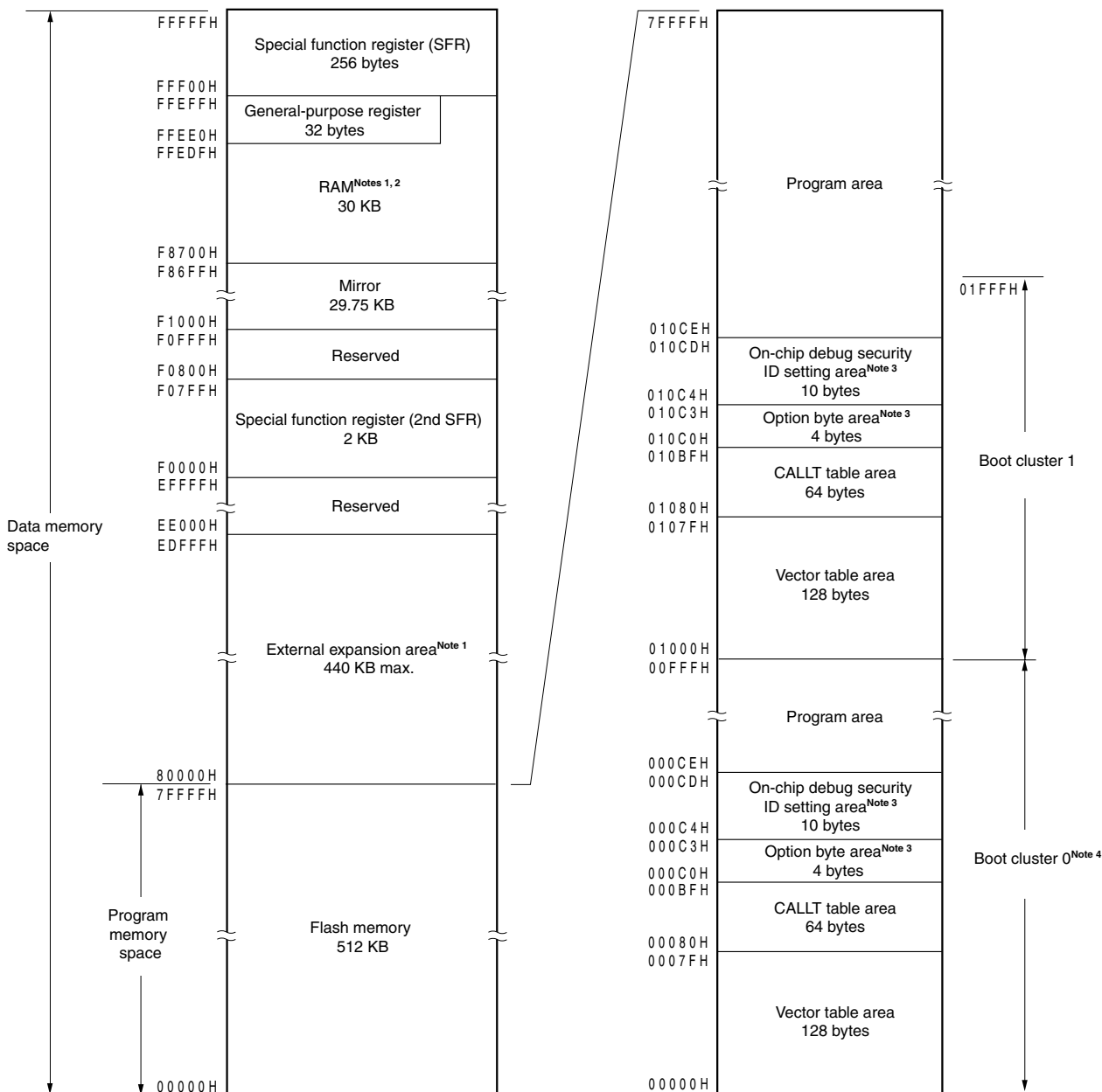
- Notes**
- Instructions can be executed from the RAM area excluding the general-purpose register area, and from the external expansion area.
  - When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.  
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.
  - Writing boot cluster 0 can be prohibited depending on the setting of security (see **25.7 Security Setting**).

Figure 3-4. Memory Map ( $\mu$ PD78F1177, 78F1177A)



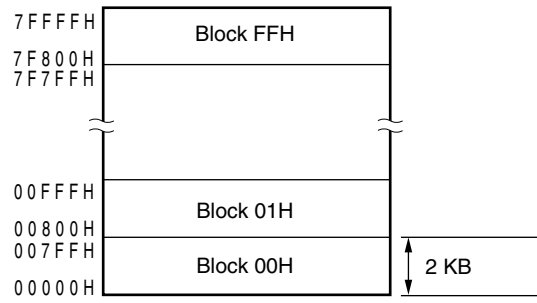
- Notes**
- Instructions can be executed from the RAM area excluding the general-purpose register area, and from the external expansion area.
  - When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.  
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.
  - Writing boot cluster 0 can be prohibited depending on the setting of security (see **25.7 Security Setting**).

Figure 3-5. Memory Map ( $\mu$ PD78F1178, 78F1178A)



- Notes**
- Instructions can be executed from the RAM area excluding the general-purpose register area, and from the external expansion area.
  - Use of the area F8700H to F8EFFH is prohibited when using the self-programming function, since this area is used for self-programming library.
  - When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.  
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.
  - Writing boot cluster 0 can be prohibited depending on the setting of security (see **25.7 Security Setting**).

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



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

**Table 3-1. Correspondence Between Address Values and Block Numbers in Flash Memory (1/2)**

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
00000H to 007FFH	00H	10000H to 107FFH	20H	20000H to 207FFH	40H	30000H to 307FFH	60H
00800H to 00FFFH	01H	10800H to 10FFFH	21H	20800H to 20FFFH	41H	30800H to 30FFFH	61H
01000H to 017FFH	02H	11000H to 117FFH	22H	21000H to 217FFH	42H	31000H to 317FFH	62H
01800H to 01FFFH	03H	11800H to 11FFFH	23H	21800H to 21FFFH	43H	31800H to 31FFFH	63H
02000H to 027FFH	04H	12000H to 127FFH	24H	22000H to 227FFH	44H	32000H to 327FFH	64H
02800H to 02FFFH	05H	12800H to 12FFFH	25H	22800H to 22FFFH	45H	32800H to 32FFFH	65H
03000H to 037FFH	06H	13000H to 137FFH	26H	23000H to 237FFH	46H	33000H to 337FFH	66H
03800H to 03FFFH	07H	13800H to 13FFFH	27H	23800H to 23FFFH	47H	33800H to 33FFFH	67H
04000H to 047FFH	08H	14000H to 147FFH	28H	24000H to 247FFH	48H	34000H to 347FFH	68H
04800H to 04FFFH	09H	14800H to 14FFFH	29H	24800H to 24FFFH	49H	34800H to 34FFFH	69H
05000H to 057FFH	0AH	15000H to 157FFH	2AH	25000H to 257FFH	4AH	35000H to 357FFH	6AH
05800H to 05FFFH	0BH	15800H to 15FFFH	2BH	25800H to 25FFFH	4BH	35800H to 35FFFH	6BH
06000H to 067FFH	0CH	16000H to 167FFH	2CH	26000H to 267FFH	4CH	36000H to 367FFH	6CH
06800H to 06FFFH	0DH	16800H to 16FFFH	2DH	26800H to 26FFFH	4DH	36800H to 36FFFH	6DH
07000H to 077FFH	0EH	17000H to 177FFH	2EH	27000H to 277FFH	4EH	37000H to 377FFH	6EH
07800H to 07FFFH	0FH	17800H to 17FFFH	2FH	27800H to 27FFFH	4FH	37800H to 37FFFH	6FH
08000H to 087FFH	10H	18000H to 187FFH	30H	28000H to 287FFH	50H	38000H to 387FFH	70H
08800H to 08FFFH	11H	18800H to 18FFFH	31H	28800H to 28FFFH	51H	38800H to 38FFFH	71H
09000H to 097FFH	12H	19000H to 197FFH	32H	29000H to 297FFH	52H	39000H to 397FFH	72H
09800H to 09FFFH	13H	19800H to 19FFFH	33H	29800H to 29FFFH	53H	39800H to 39FFFH	73H
0A000H to 0A7FFH	14H	1A000H to 1A7FFH	34H	2A000H to 2A7FFH	54H	3A000H to 3A7FFH	74H
0A800H to 0AFFFH	15H	1A800H to 1AFFFH	35H	2A800H to 2AFFFH	55H	3A800H to 3AFFFH	75H
0B000H to 0B7FFH	16H	1B000H to 1B7FFH	36H	2B000H to 2B7FFH	56H	3B000H to 3B7FFH	76H
0B800H to 0BFFFH	17H	1B800H to 1BFFFH	37H	2B800H to 2BFFFH	57H	3B800H to 3BFFFH	77H
0C000H to 0C7FFH	18H	1C000H to 1C7FFH	38H	2C000H to 2C7FFH	58H	3C000H to 3C7FFH	78H
0C800H to 0CFFFH	19H	1C800H to 1CFFFH	39H	2C800H to 2CFFFH	59H	3C800H to 3CFFFH	79H
0D000H to 0D7FFH	1AH	1D000H to 1D7FFH	3AH	2D000H to 2D7FFH	5AH	3D000H to 3D7FFH	7AH
0D800H to 0DFFFH	1BH	1D800H to 1DFFFH	3BH	2D800H to 2DFFFH	5BH	3D800H to 3DFFFH	7BH
0E000H to 0E7FFH	1CH	1E000H to 1E7FFH	3CH	2E000H to 2E7FFH	5CH	3E000H to 3E7FFH	7CH
0E800H to 0EFFFH	1DH	1E800H to 1EFFFH	3DH	2E800H to 2EFFFH	5DH	3E800H to 3EFFFH	7DH
0F000H to 0F7FFH	1EH	1F000H to 1F7FFH	3EH	2F000H to 2F7FFH	5EH	3F000H to 3F7FFH	7EH
0F800H to 0FFFFH	1FH	1F800H to 1FFFFH	3FH	2F800H to 2FFFFH	5FH	3F800H to 3FFFFH	7FH

**Remark**  $\mu$ PD78F1174, 78F1174A: Block numbers 00H to 3FH  
 $\mu$ PD78F1175, 78F1175A: Block numbers 00H to 5FH  
 $\mu$ PD78F1176, 78F1176A: Block numbers 00H to 7FH  
 $\mu$ PD78F1177, 78F1177A: Block numbers 00H to BFH  
 $\mu$ PD78F1178, 78F1178A: Block numbers 00H to FFH

Table 3-1. Correspondence Between Address Values and Block Numbers in Flash Memory (2/2)

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
40000H to 407FFH	80H	50000H to 507FFH	A0H	60000H to 607FFH	C0H	70000H to 707FFH	E0H
40800H to 40FFFH	81H	50800H to 50FFFH	A1H	60800H to 60FFFH	C1H	70800H to 70FFFH	E1H
41000H to 417FFH	82H	51000H to 517FFH	A2H	61000H to 617FFH	C2H	71000H to 717FFH	E2H
41800H to 41FFFH	83H	51800H to 51FFFH	A3H	61800H to 61FFFH	C3H	71800H to 71FFFH	E3H
42000H to 427FFH	84H	52000H to 527FFH	A4H	62000H to 627FFH	C4H	72000H to 727FFH	E4H
42800H to 42FFFH	85H	52800H to 52FFFH	A5H	62800H to 62FFFH	C5H	72800H to 72FFFH	E5H
43000H to 437FFH	86H	53000H to 537FFH	A6H	63000H to 637FFH	C6H	73000H to 737FFH	E6H
43800H to 43FFFH	87H	53800H to 53FFFH	A7H	63800H to 63FFFH	C7H	73800H to 73FFFH	E7H
44000H to 447FFH	88H	54000H to 547FFH	A8H	64000H to 647FFH	C8H	74000H to 747FFH	E8H
44800H to 44FFFH	89H	54800H to 54FFFH	A9H	64800H to 64FFFH	C9H	74800H to 74FFFH	E9H
45000H to 457FFH	8AH	55000H to 557FFH	AAH	65000H to 657FFH	CAH	75000H to 757FFH	EAH
45800H to 45FFFH	8BH	55800H to 55FFFH	ABH	65800H to 65FFFH	CBH	75800H to 75FFFH	EBH
46000H to 467FFH	8CH	56000H to 567FFH	ACH	66000H to 667FFH	CCH	76000H to 767FFH	ECH
46800H to 46FFFH	8DH	56800H to 56FFFH	ADH	66800H to 66FFFH	CDH	76800H to 76FFFH	EDH
47000H to 477FFH	8EH	57000H to 577FFH	AEH	67000H to 677FFH	CEH	77000H to 777FFH	EEH
47800H to 47FFFH	8FH	57800H to 57FFFH	AFH	67800H to 67FFFH	CFH	77800H to 77FFFH	EFH
48000H to 487FFH	90H	58000H to 587FFH	B0H	68000H to 687FFH	D0H	78000H to 787FFH	F0H
48800H to 48FFFH	91H	58800H to 58FFFH	B1H	68800H to 68FFFH	D1H	78800H to 78FFFH	F1H
49000H to 497FFH	92H	59000H to 597FFH	B2H	69000H to 697FFH	D2H	79000H to 797FFH	F2H
49800H to 49FFFH	93H	59800H to 59FFFH	B3H	69800H to 69FFFH	D3H	79800H to 79FFFH	F3H
4A000H to 4A7FFH	94H	5A000H to 5A7FFH	B4H	6A000H to 6A7FFH	D4H	7A000H to 7A7FFH	F4H
4A800H to 4AFFFH	95H	5A800H to 5AFFFH	B5H	6A800H to 6AFFFH	D5H	7A800H to 7AFFFH	F5H
4B000H to 4B7FFH	96H	5B000H to 5B7FFH	B6H	6B000H to 6B7FFH	D6H	7B000H to 7B7FFH	F6H
4B800H to 4BFFFH	97H	5B800H to 5BFFFH	B7H	6B800H to 6BFFFH	D7H	7B800H to 7BFFFH	F7H
4C000H to 4C7FFH	98H	5C000H to 5C7FFH	B8H	6C000H to 6C7FFH	D8H	7C000H to 7C7FFH	F8H
4C800H to 4CFFFH	99H	5C800H to 5CFFFH	B9H	6C800H to 6CFFFH	D9H	7C800H to 7CFFFH	F9H
4D000H to 4D7FFH	9AH	5D000H to 5D7FFH	BAH	6D000H to 6D7FFH	DAH	7D000H to 7D7FFH	FAH
4D800H to 4DFFFH	9BH	5D800H to 5DFFFH	BBH	6D800H to 6DFFFH	DBH	7D800H to 7DFFFH	FBH
4E000H to 4E7FFH	9CH	5E000H to 5E7FFH	BCH	6E000H to 6E7FFH	DCH	7E000H to 7E7FFH	FCH
4E800H to 4EFFFH	9DH	5E800H to 5EFFFH	BDH	6E800H to 6EFFFH	DDH	7E800H to 7EFFFH	FDH
4F000H to 4F7FFH	9EH	5F000H to 5F7FFH	BEH	6F000H to 6F7FFH	DEH	7F000H to 7F7FFH	FEH
4F800H to 4FFFFH	9FH	5F800H to 5FFFFH	BFH	6F800H to 6FFFFH	DFH	7F800H to 7FFFFH	FFH

**Remark**  $\mu$ PD78F1174, 78F1174A: Block numbers 00H to 3FH  
 $\mu$ PD78F1175, 78F1175A: Block numbers 00H to 5FH  
 $\mu$ PD78F1176, 78F1176A: Block numbers 00H to 7FH  
 $\mu$ PD78F1177, 78F1177A: Block numbers 00H to BFH  
 $\mu$ PD78F1178, 78F1178A: Block numbers 00H to FFH



### 3.1.1 Internal program memory space

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

78K0R/KH3 products incorporate internal ROM (flash memory), as shown below.

**Table 3-2. Internal ROM Capacity**

Part Number	Internal ROM	
	Structure	Capacity
$\mu$ PD78F1174, 78F1174A	Flash memory	131072 $\times$ 8 bits (00000H to 1FFFFH)
$\mu$ PD78F1175, 78F1175A		196608 $\times$ 8 bits (00000H to 2FFFFH)
$\mu$ PD78F1176, 78F1176A		262144 $\times$ 8 bits (00000H to 3FFFFH)
$\mu$ PD78F1177, 78F1177A		393216 $\times$ 8 bits (00000H to 5FFFFH)
$\mu$ PD78F1178, 78F1178A		524288 $\times$ 8 bits (00000H to 7FFFFH)

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

#### (1) Vector table area

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

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

Table 3-3. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
00000H	RESET input, POC, LVI, WDT, TRAP	00030H	INTTM02
		00032H	INTTM03
00004H	INTWDTI	00034H	INTAD
00006H	INTLVI	00036H	INTRTC
00008H	INTP0	00038H	INTRTCI
0000AH	INTP1	0003AH	INTKR
0000CH	INTP2	0003CH	INTST2/INTCSI20/INTIIC20
0000EH	INTP3	0003EH	INTSR2/INTCSI21/INTIIC21
00010H	INTP4	00040H	INTSRE2
00012H	INTP5	00042H	INTTM04
00014H	INTST3	00044H	INTTM05
00016H	INTSR3	00046H	INTTM06
00018H	INTSRE3	00048H	INTTM07
0001AH	INTDMA0	0004AH	INTP6
0001CH	INTDMA1	0004CH	INTP7
0001EH	INTST0/INTCSI00	0004EH	INTP8
00020H	INTSR0/INTCSI01	00050H	INTP9
00022H	INTSRE0	00052H	INTP10
00024H	INTST1/INTCSI10/INTIIC10	00054H	INTP11
00026H	INTSR1/INTCSI11/INTIIC11	00056H	INTTM10
00028H	INTSRE1	00058H	INTTM11
0002AH	INTIIC0	0005AH	INTTM12
0002CH	INTTM00	0005CH	INTTM13
0002EH	INTTM01	0007EH	BRK

**(2) CALLT instruction table area**

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

To use the boot swap function, set a CALLT instruction table also at 01080H to 010BFH.

**(3) Option byte area**

A 4-byte area of 000C0H to 000C3H can be used as an option byte area. Set the option byte at 010C0H to 010C3H when the boot swap is used. For details, see **CHAPTER 24 OPTION BYTE**.

**(4) On-chip debug security ID setting area**

A 10-byte area of 000C4H to 000CDH and 010C4H to 010CDH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 000C4H to 000CDH when the boot swap is not used and at 000C4H to 000CDH and 010C4H to 010CDH when the boot swap is used. For details, see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.

**3.1.2 Mirror area**

The 78K0R/KH3 mirrors the data flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, to F0000H to FFFFFH (the data flash area to be mirrored is set by the processor mode control register (PMC)).

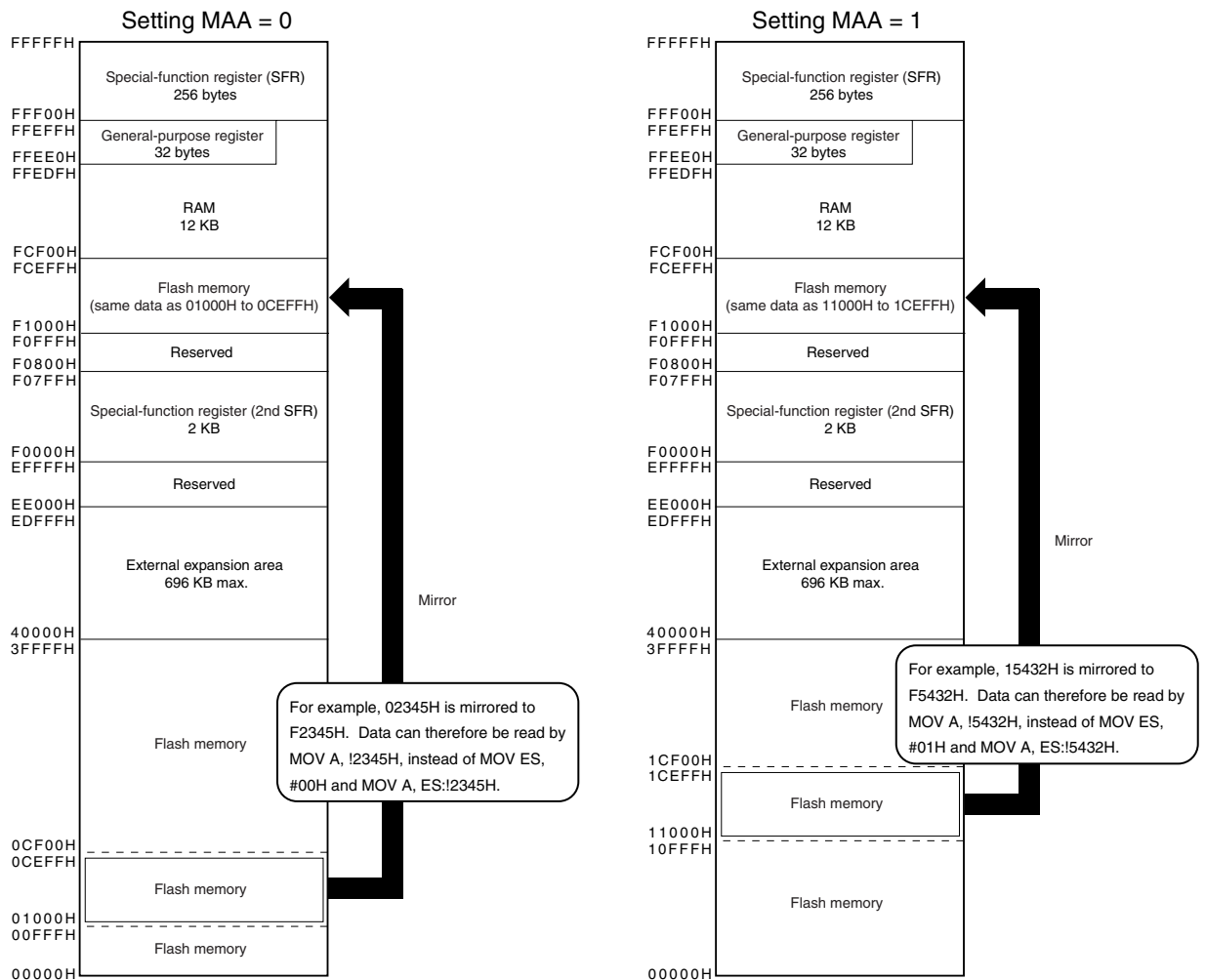
By reading data from F0000H to FFFFFH, an instruction that does not have the ES registers as an operand can be used, and thus the contents of the data flash can be read with the shorter code. However, the data flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

See **3.1 Memory Space** for the mirror area of each product.

The mirror area can only be read and no instruction can be fetched from this area.

The following show examples.

**Example  $\mu$ PD78F1176, 78F1176A (Flash memory: 256 KB, RAM: 12 KB)**



**Remark** MAA: Bit 0 of the processor mode control register (PMC).

PMC register is described below.

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

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

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

Reset signal generation sets this register to 00H.

**Figure 3-6. Format of Configuration of Processor Mode Control Register (PMC)**

Address: FFFFEH After reset: 00H R/W

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

MAA	Selection of flash memory space for mirroring to area from F0000H to FFFFFH
0	00000H to 0FFFFH is mirrored to F0000H to FFFFFH
1	10000H to 1FFFFH is mirrored to F0000H to FFFFFH

**Cautions** 1. Set PMC only once during the initial settings prior to operating the DMA controller. Rewriting PMC other than during the initial settings is prohibited.

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

**3.1.3 Internal data memory space**

78K0R/KH3 products incorporate the following RAMs.

**Table 3-4. Internal RAM Capacity**

Part Number	Internal RAM
$\mu$ PD78F1174, 78F1174A	8192 $\times$ 8 bits (FDF00H to FFEFFH)
$\mu$ PD78F1175, 78F1175A	10240 $\times$ 8 bits (FD700H to FFEFFH)
$\mu$ PD78F1176, 78F1176A	12288 $\times$ 8 bits (FCF00H to FFEFFH)
$\mu$ PD78F1177, 78F1177A	24576 $\times$ 8 bits (F9F00H to FFEFFH)
$\mu$ PD78F1178, 78F1178A	30720 $\times$ 8 bits (F8700H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are written and executed. Four general-purpose register banks consisting of eight 8-bit registers per bank are assigned to the 32-byte area of FFEE0H to FFEFFH of the internal RAM area. However, instructions cannot be executed by using general-purpose registers.

The internal RAM is used as a stack memory.

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

2. While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory. Furthermore, the areas of F8700H to F8EFFH cannot be used with the  $\mu$ PD78F1178 and 78F1178A, respectively.

### 3.1.4 Special function register (SFR) area

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

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

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

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

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

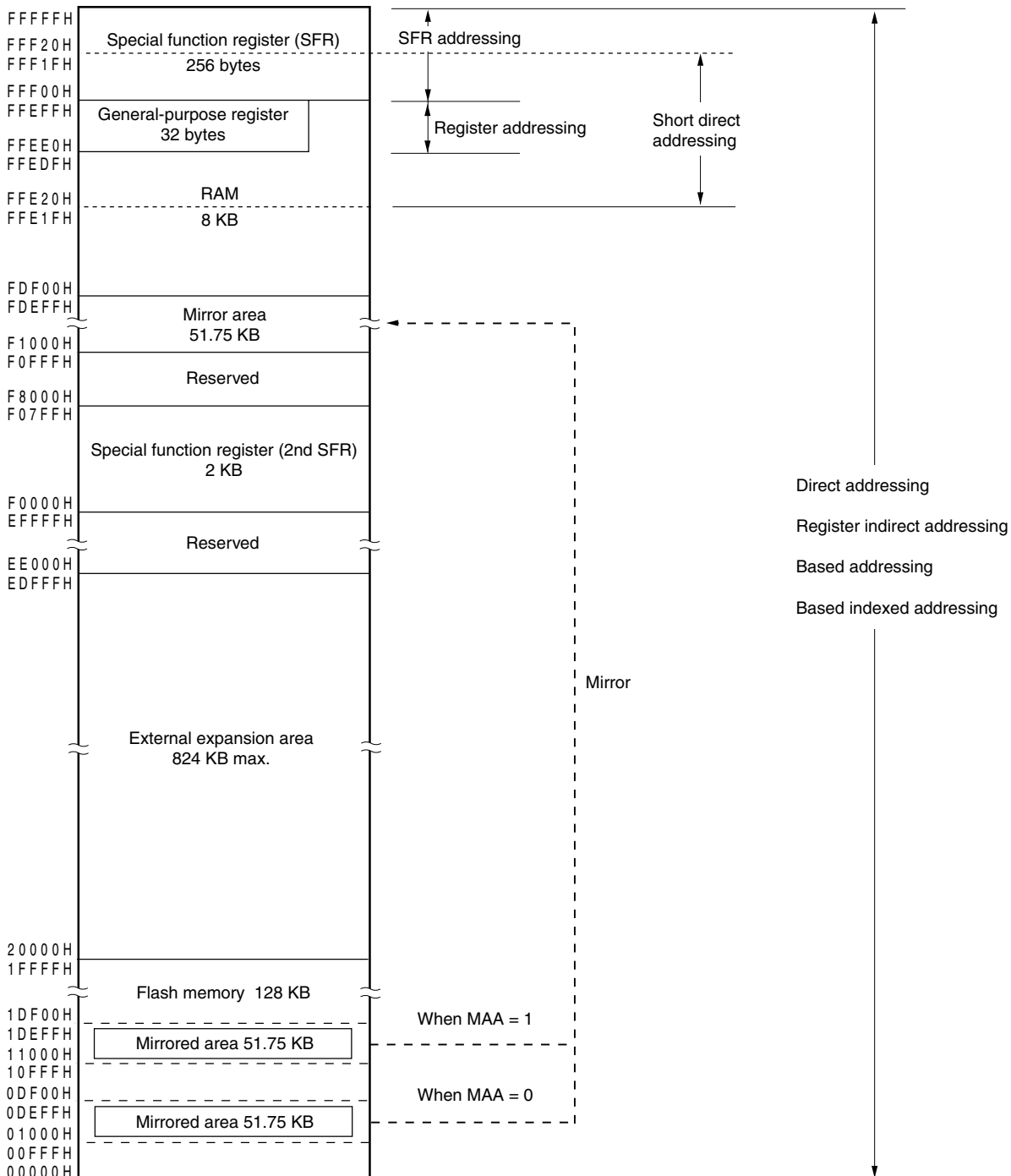
**Caution** Do not access addresses to which 2nd SFRs are not assigned.

### 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.

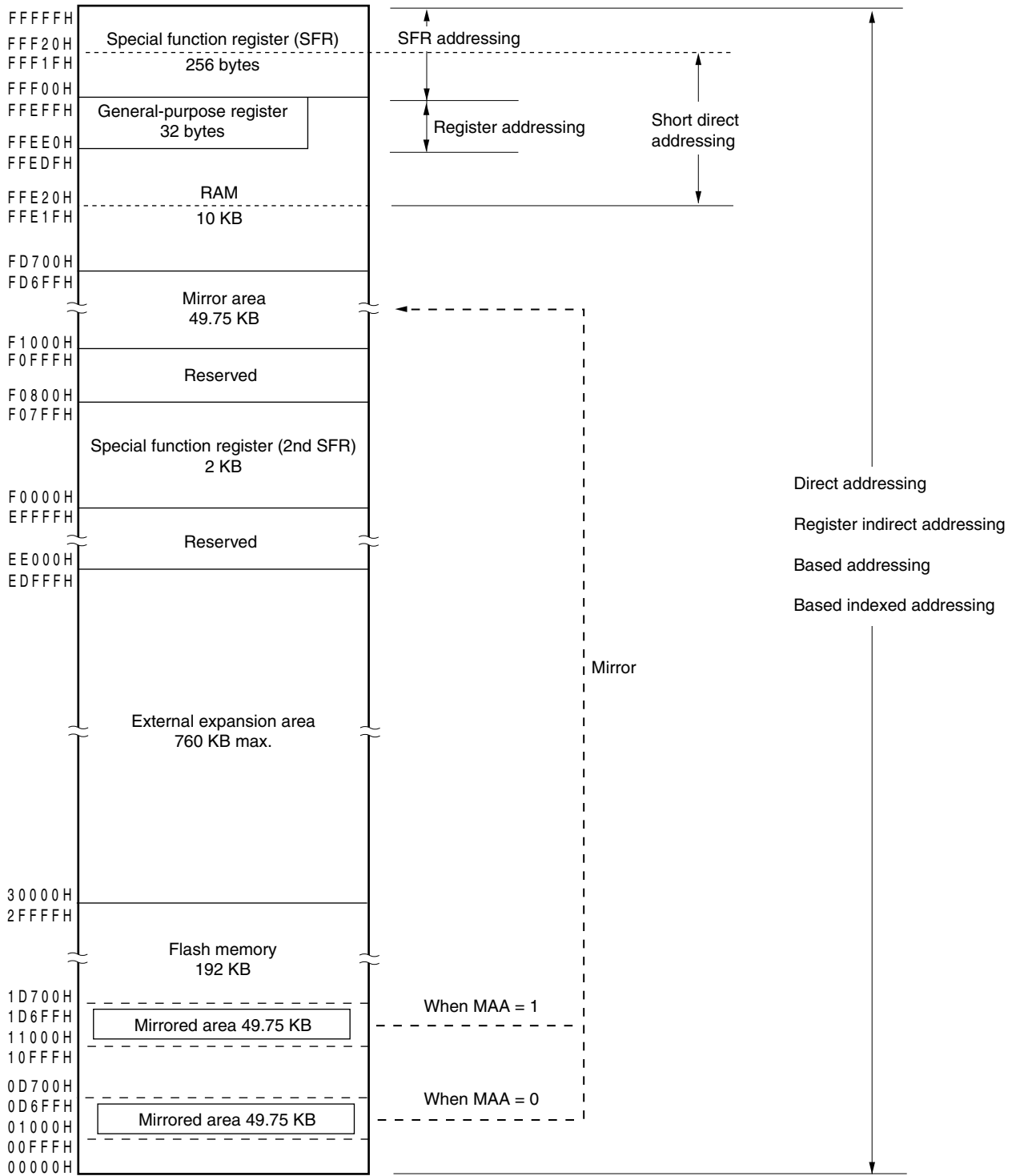
Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the 78K0R/KH3, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Figures 3-7 to 3-11 show correspondence between data memory and addressing.

<R> **Figure 3-7. Correspondence Between Data Memory and Addressing ( $\mu$ PD78F1174, 78F1174A)**

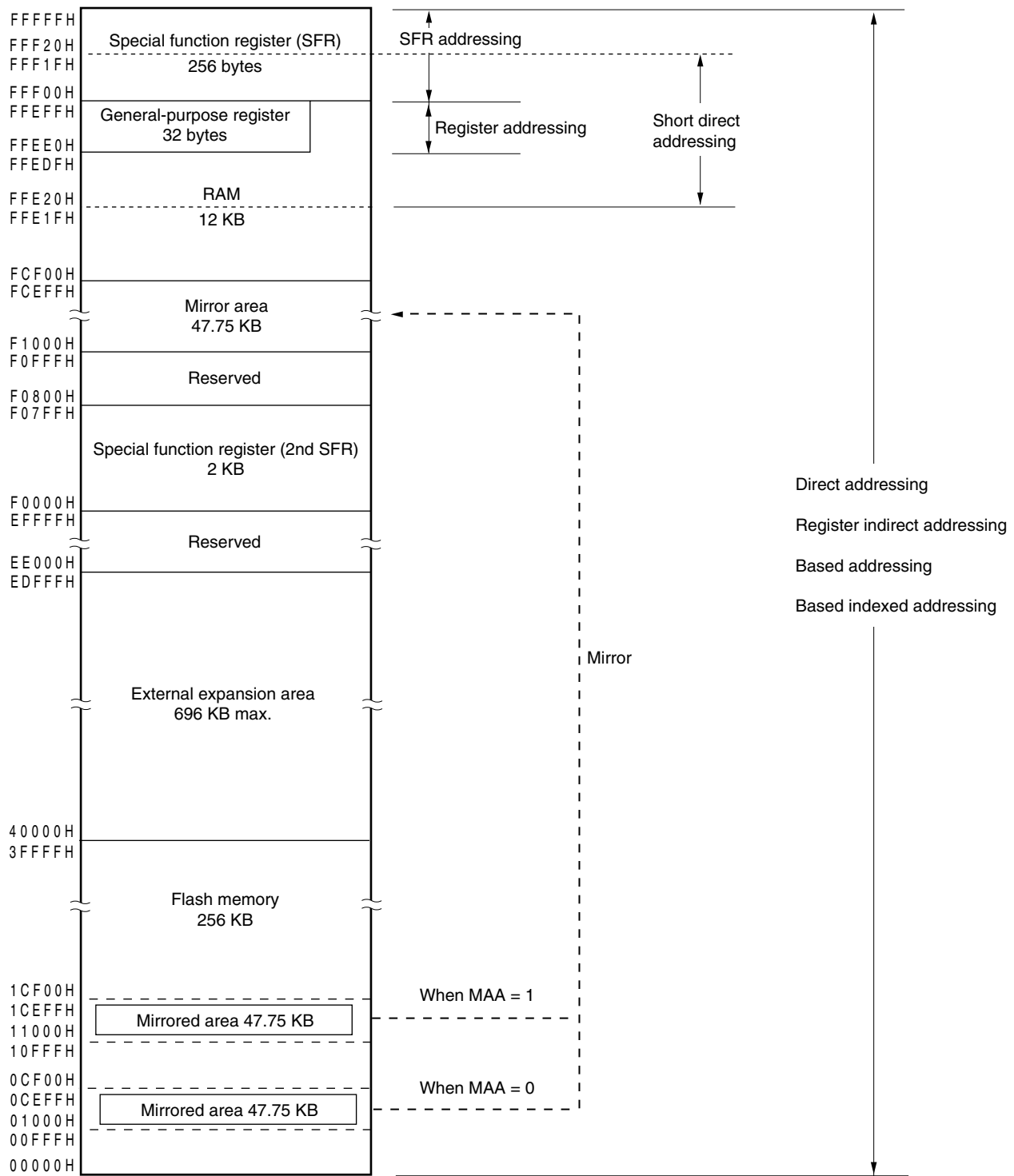


<R>

Figure 3-8. Correspondence Between Data Memory and Addressing ( $\mu$ PD78F1175, 78F1175A)



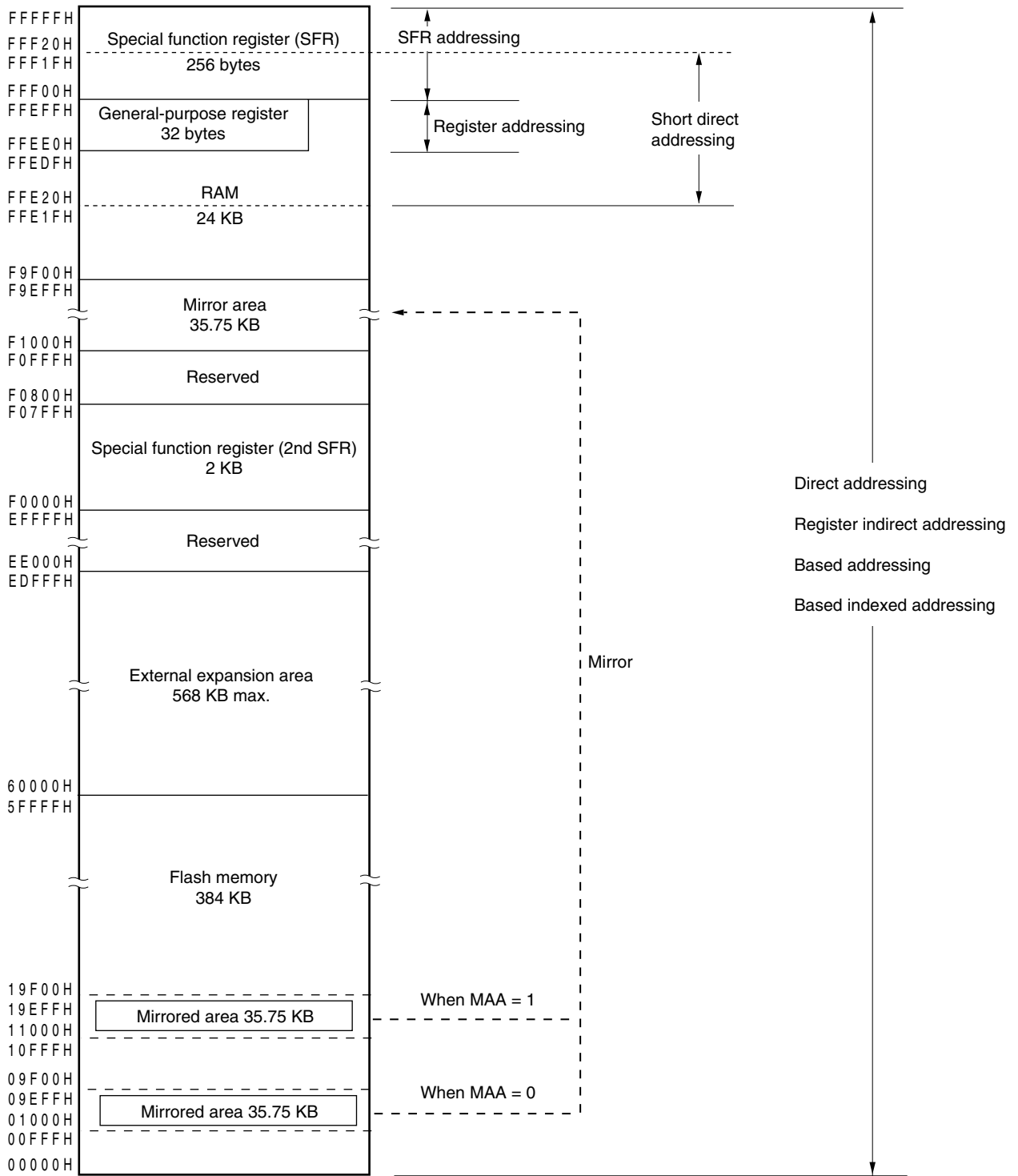
<R> **Figure 3-9. Correspondence Between Data Memory and Addressing ( $\mu$ PD78F1176, 78F1176A)**



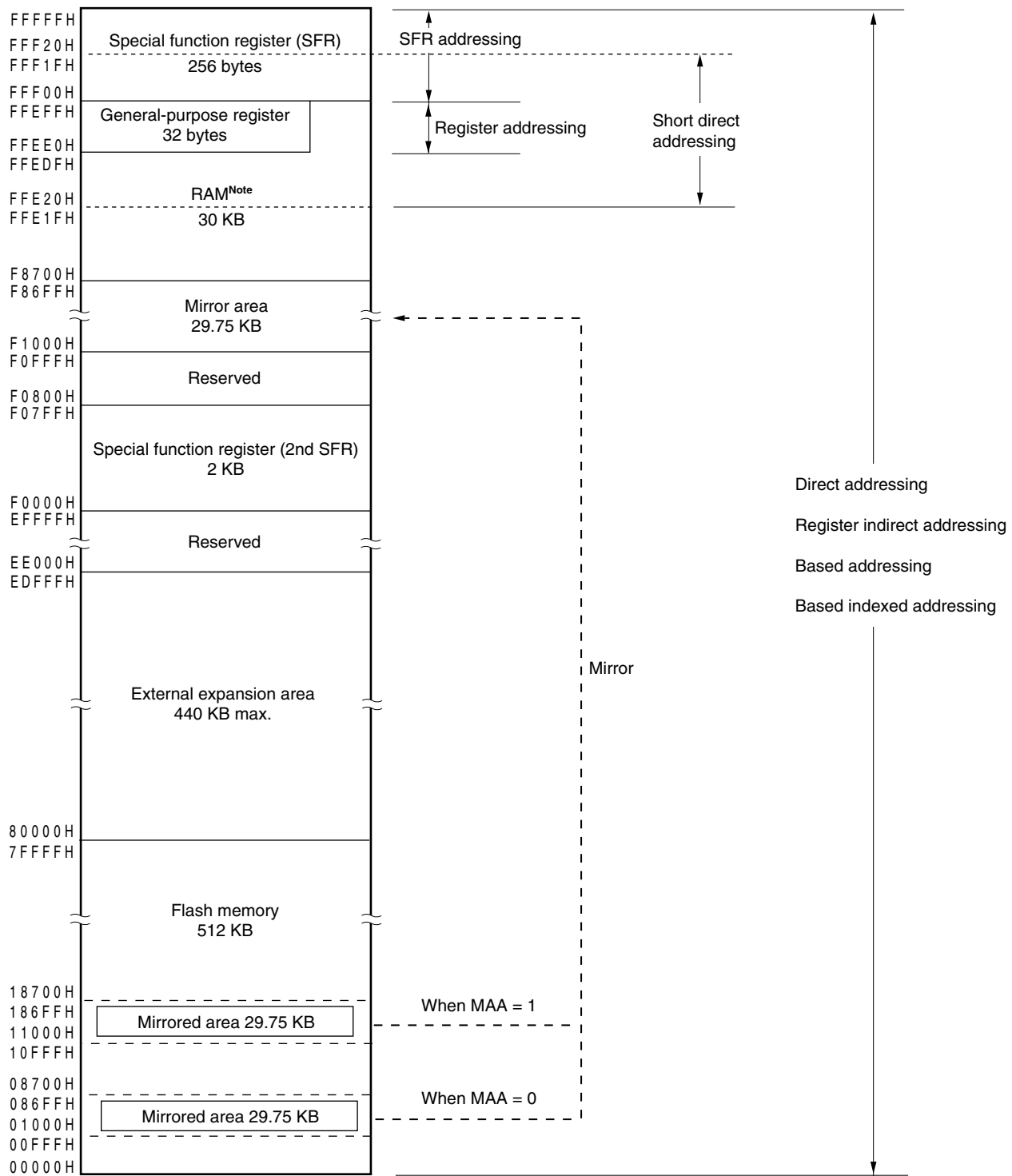


<R>

Figure 3-10. Correspondence Between Data Memory and Addressing ( $\mu$ PD78F1177, 78F1177A)



<R> **Figure 3-11. Correspondence Between Data Memory and Addressing ( $\mu$ PD78F1178, 78F1178A)**



**Note** Use of the area F8700H to F86FFH is prohibited when using the self-programming function. Since this area is used for self-programming library.

## 3.2 Processor Registers

The 78K0R/KH3 products incorporate the following processor registers.

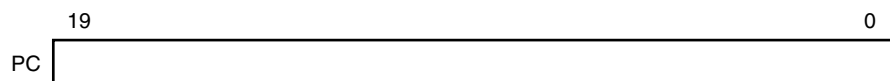
### 3.2.1 Control registers

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 program counter.

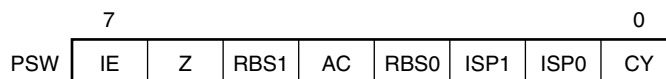
**Figure 3-12. Format of Program Counter**



#### (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 vector interrupt request acknowledgment or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions. Reset signal generation sets PSW to 06H.

**Figure 3-13. Format of Program Status Word**



##### (a) Interrupt enable flag (IE)

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

##### (b) Zero flag (Z)

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

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

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

**(d) Auxiliary carry flag (AC)**

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

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

This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 by a priority specification flag register (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see 17.3 (3)) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

**Remark** n = 0, 1

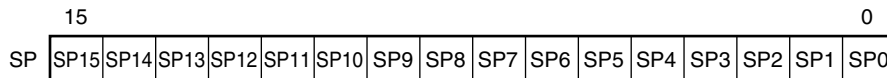
**(f) Carry flag (CY)**

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

**(3) Stack pointer (SP)**

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

**Figure 3-14. Format of Stack Pointer**



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 3-15.

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

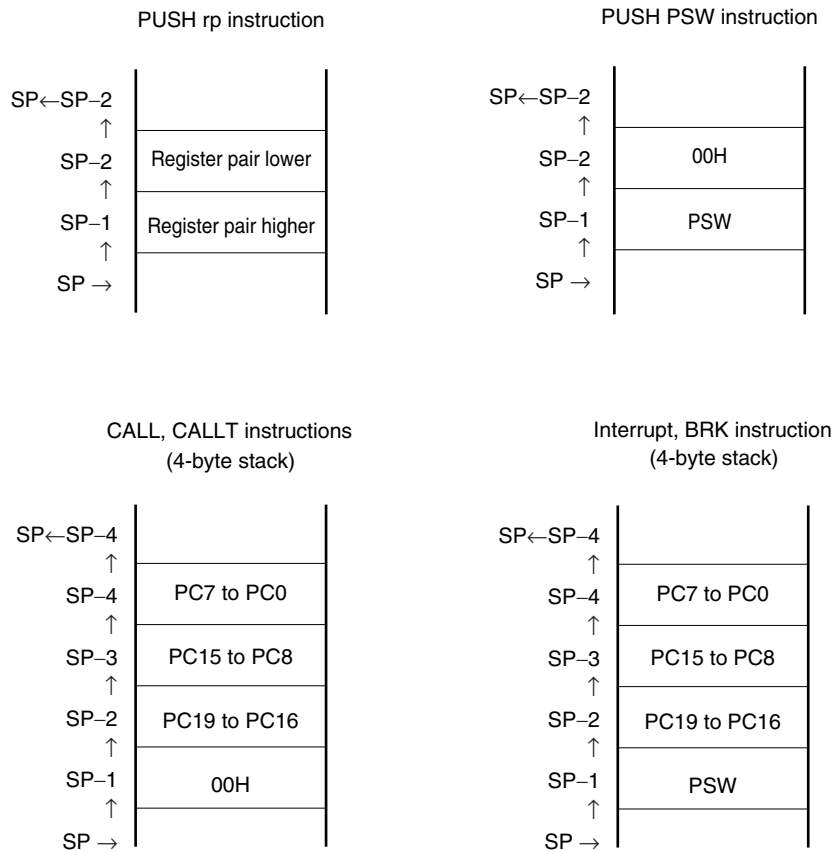
2. The values of the stack pointer must be set to even numbers. If odd numbers are specified, the least significant bit is automatically cleared to 0.

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

4. While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory. Furthermore, the areas of F8700H to F8EFFH also cannot be used with the  $\mu$ PD78F1178 and 78F1178A, respectively.

<R>

Figure 3-15. Data to Be Saved to Stack Memory



### 3.2.2 General-purpose registers

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

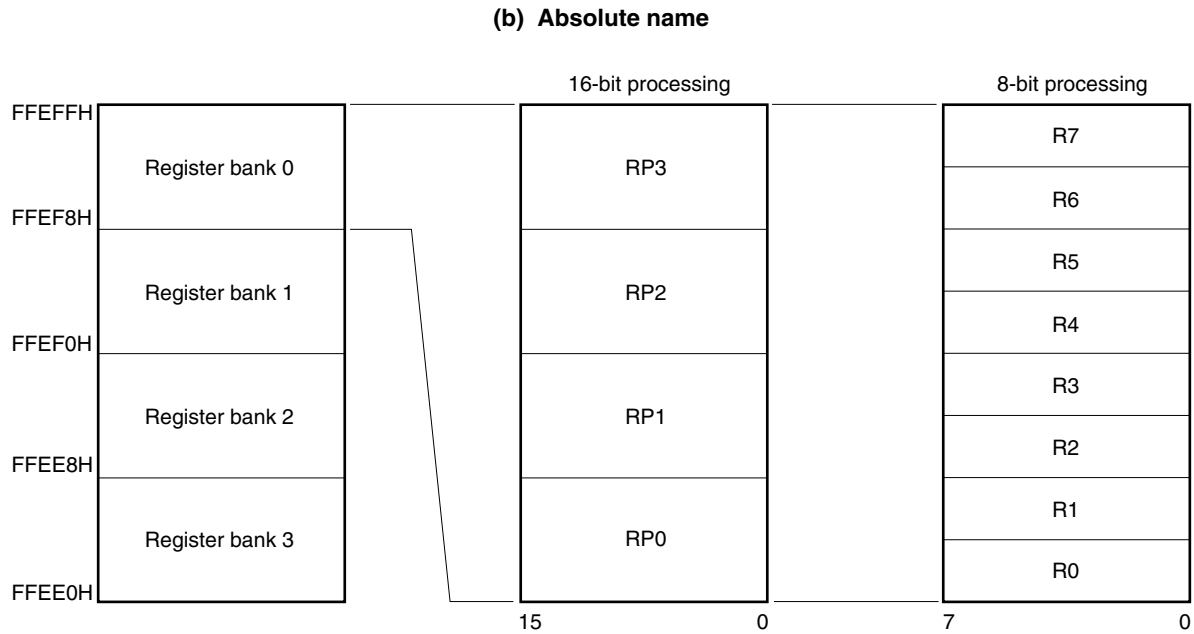
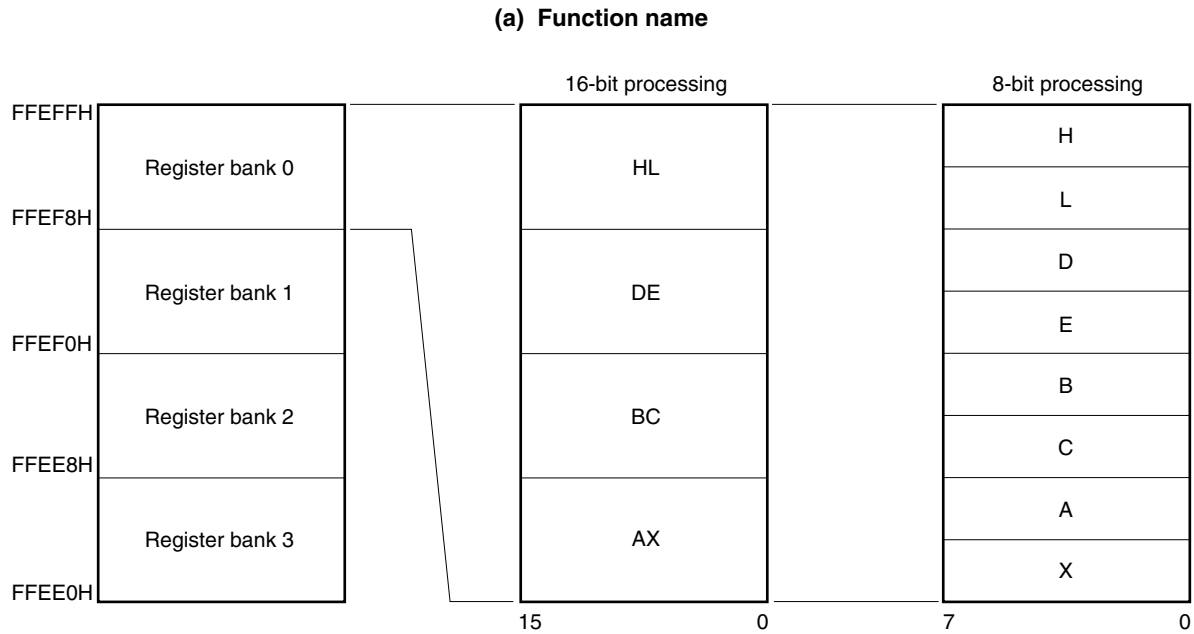
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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

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

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

Figure 3-16. Configuration of General-Purpose Registers



### 3.2.3 ES and CS registers

The ES register is used for data access and the CS register is used to specify the higher address when a branch instruction is executed.

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

**Figure 3-17. Configuration of ES and CS Registers**

	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0
CS	0	0	0	0	CS3	CP2	CP1	CP0

### 3.2.4 Special function registers (SFRs)

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

SFRs are allocated to the FFF00H to FFFFFH area.

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

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation  
Describe 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.
- 16-bit manipulation  
Describe the symbol reserved by the assembler for the 16-bit manipulation instruction operand (sfrp). When specifying an address, describe an even address.

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

- Symbol  
Symbol indicating the address of a special function register. It is a reserved word in the RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R. When using the RA78K0R, ID78K0R-QB, and SM+ for 78K0R, symbols can be written as an instruction operand.
- R/W  
Indicates whether the corresponding SFR can be read or written.  
R/W: Read/write enable  
R: Read only  
W: Write only
- Manipulable bit units  
“√/” indicates the manipulable bit unit (1, 8, or 16). “-” indicates a bit unit for which manipulation is not possible.
- After reset  
Indicates each register status upon reset signal generation.

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

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



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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF00H	Port register 0	P0		R/W	√	√	–	00H
FFF01H	Port register 1	P1		R/W	√	√	–	00H
FFF02H	Port register 2	P2		R/W	√	√	–	00H
FFF03H	Port register 3	P3		R/W	√	√	–	00H
FFF04H	Port register 4	P4		R/W	√	√	–	00H
FFF05H	Port register 5	P5		R/W	√	√	–	00H
FFF06H	Port register 6	P6		R/W	√	√	–	00H
FFF07H	Port register 7	P7		R/W	√	√	–	00H
FFF08H	Port register 8	P8		R/W	√	√	–	00H
FFF09H	Port register 9	P9		R/W	√	√	–	00H
FFF0BH	Port register 11	P11		R/W	√	√	–	00H
FFF0CH	Port register 12	P12		R/W	√	√	–	Undefined
FFF0DH	Port register 13	P13		R/W	√	√	–	00H
FFF0EH	Port register 14	P14		R/W	√	√	–	00H
FFF0FH	Port register 15	P15		R/W	√	√	–	00H
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	–	√	√	0000H
FFF11H		–			–	–		
FFF12H	Serial data register 01	RXD0/ SIO01	SDR01	R/W	–	√	√	0000H
FFF13H		–			–	–		
FFF14H	Serial data register 12	TXD3	SDR12	R/W	–	√	√	0000H
FFF15H		–			–	–		
FFF16H	Serial data register 13	RXD3	SDR13	R/W	–	√	√	0000H
FFF17H		–			–	–		
FFF18H	Timer data register 00	TDR00		R/W	–	–	√	0000H
FFF19H					–	–	–	
FFF1AH	Timer data register 01	TDR01		R/W	–	–	√	0000H
FFF1BH					–	–	–	
FFF1CH	8-bit D/A conversion value setting register 0	DACS0		R/W	√	√	–	00H
FFF1DH	8-bit D/A conversion value setting register 1	DACS1		R/W	√	√	–	00H
FFF1EH	10-bit A/D conversion result register	ADCR		R	–	–	√	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	–	√	–	00H
FFF20H	Port mode register 0	PM0		R/W	√	√	–	FFH
FFF21H	Port mode register 1	PM1		R/W	√	√	–	FFH
FFF22H	Port mode register 2	PM2		R/W	√	√	–	FFH
FFF23H	Port mode register 3	PM3		R/W	√	√	–	FFH
FFF24H	Port mode register 4	PM4		R/W	√	√	–	FFH
FFF25H	Port mode register 5	PM5		R/W	√	√	–	FFH
FFF26H	Port mode register 6	PM6		R/W	√	√	–	FFH
FFF27H	Port mode register 7	PM7		R/W	√	√	–	FFH
FFF28H	Port mode register 8	PM8		R/W	√	√	–	FFH
FFF29H	Port mode register 9	PM9		R/W	√	√	–	FFH
FFF2BH	Port mode register 11	PM11		R/W	√	√	–	FFH
FFF2CH	Port mode register 12	PM12		R/W	√	√	–	FFH

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF2DH	Port mode register 13	PM13		R/W	√	√	–	FEH
FFF2EH	Port mode register 14	PM14		R/W	√	√	–	FFH
FFF2FH	Port mode register 15	PM15		R/W	√	√	–	FFH
FFF30H	A/D converter mode register	ADM		R/W	√	√	–	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	–	00H
FFF32H	D/A converter mode register	DAM		R/W	√	√	–	00H
FFF34H	Port register 16	P16		R/W	√	√	–	00H
FFF35H	Port mode register 16	PM16		R/W	√	√	–	FFH
FFF37H	Key return mode register	KRM		R/W	√	√	–	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	–	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	–	00H
FFF3AH	External interrupt rising edge enable register 1	EGP1		R/W	√	√	–	00H
FFF3BH	External interrupt falling edge enable register 1	EGN1		R/W	√	√	–	00H
FFF3CH	Input switch control register	ISC		R/W	√	√	–	00H
FFF3EH	Timer input select register 0	TIS0		R/W	√	√	–	00H
FFF3FH	Timer input select register 1	TIS1		R/W	√	√	–	00H
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	–	√	√	0000H
FFF45H		–			–	–		
FFF46H	Serial data register 03	RXD1/ SIO11	SDR03	R/W	–	√	√	0000H
FFF47H		–			–	–		
FFF48H	Serial data register 10	TXD2/ SIO20	SDR10	R/W	–	√	√	0000H
FFF49H		–			–	–		
FFF4AH	Serial data register 11	RXD2/ SIO21	SDR11	R/W	–	√	√	0000H
FFF4BH		–			–	–		
FFF50H	IIC shift register 0	IIC0		R/W	–	√	–	00H
FFF51H	IIC flag register 0	IICF0		R/W	√	√	–	00H
FFF52H	IIC control register 0	IICC0		R/W	√	√	–	00H
FFF53H	IIC slave address register 0	SVA0		R/W	–	√	–	00H
FFF54H	IIC clock select register 0	IICCL0		R/W	√	√	–	00H
FFF55H	IIC function expansion register 0	IICX0		R/W	√	√	–	00H
FFF56H	IIC status register 0	IICS0		R	√	√	–	00H
FFF64H	Timer data register 02	TDR02		R/W	–	–	√	0000H
FFF65H					–	–	–	
FFF66H	Timer data register 03	TDR03		R/W	–	–	√	0000H
FFF67H					–	–	–	
FFF68H	Timer data register 04	TDR04		R/W	–	–	√	0000H
FFF69H					–	–	–	
FFF6AH	Timer data register 05	TDR05		R/W	–	–	√	0000H
FFF6BH					–	–	–	
FFF6CH	Timer data register 06	TDR06		R/W	–	–	√	0000H
FFF6DH					–	–	–	
FFF6EH	Timer data register 07	TDR07		R/W	–	–	√	0000H
FFF6FH					–	–	–	

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
FFF70H	Timer data register 10	TDR10	R/W	–	–	√	0000H
FFF71H							
FFF72H	Timer data register 11	TDR11	R/W	–	–	√	0000H
FFF73H							
FFF74H	Timer data register 12	TDR12	R/W	–	–	√	0000H
FFF75H							
FFF76H	Timer data register 13	TDR13	R/W	–	–	√	0000H
FFF77H							
FFF90H	Sub-count register	RSUBC	R	–	–	√	0000H
FFF91H							
FFF92H	Second count register	SEC	R/W	–	√	–	00H
FFF93H	Minute count register	MIN	R/W	–	√	–	00H
FFF94H	Hour count register	HOUR	R/W	–	√	–	12H <sup>Note 1</sup>
FFF95H	Week count register	WEEK	R/W	–	√	–	00H
FFF96H	Day count register	DAY	R/W	–	√	–	01H
FFF97H	Month count register	MONTH	R/W	–	√	–	01H
FFF98H	Year count register	YEAR	R/W	–	√	–	00H
FFF99H	Watch error correction register	SUBCUD	R/W	–	√	–	00H
FFF9AH	Alarm minute register	ALARMWM	R/W	–	√	–	00H
FFF9BH	Alarm hour register	ALARMWH	R/W	–	√	–	12H
FFF9CH	Alarm week register	ALARMWW	R/W	–	√	–	00H
FFF9DH	Real-time counter control register 0	RTCC0	R/W	√	√	–	00H
FFF9EH	Real-time counter control register 1	RTCC1	R/W	√	√	–	00H
FFF9FH	Real-time counter control register 2	RTCC2	R/W	√	√	–	00H
FFFA0H	Clock operation mode control register	CMC	R/W	–	√	–	00H
FFFA1H	Clock operation status control register	CSC	R/W	√	√	–	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC	R	√	√	–	00H
FFFA3H	Oscillation stabilization time select register	OSTS	R/W	–	√	–	07H
FFFA4H	System clock control register	CKC	R/W	√	√	–	09H
FFFA5H	Clock output select register 0	CKS0	R/W	√	√	–	00H
FFFA6H	Clock output select register 1	CKS1	R/W	√	√	–	00H
FFFA8H	Reset control flag register	RESF	R	–	√	–	00H <sup>Note 2</sup>
FFFA9H	Low-voltage detection register	LVIM	R/W	√	√	–	00H <sup>Note 3</sup>
FFFAAH	Low-voltage detection level select register	LVIS	R/W	√	√	–	0EH <sup>Note 4</sup>
FFFABH	Watchdog timer enable register	WDTE	R/W	–	√	–	1A/9A <sup>Note 5</sup>
FFFACH	–	TTBLH <sup>Note 6</sup>	–	–	–	–	Undefined
FFFADH	–						
FFFAEH	–	TTBL <sup>Note 6</sup>	–	–	–	–	Undefined
FFF AFH	–						

- Notes**
1. The value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.
  2. The reset value of RESF varies depending on the reset source.
  3. The reset value of LVIM varies depending on the reset source and the setting of the option byte.
  4. The reset value of LVIS varies depending on the reset source.
  5. The reset value of WDTE is determined by the setting of the option byte.
  6. This SFR cannot be used by the user, so do not operate it directly.

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFB0H	DMA SFR address register 0	DSA0		R/W	–	√	–	00H
FFFB1H	DMA SFR address register 1	DSA1		R/W	–	√	–	00H
FFFB2H	DMA RAM address register 0L	DRA0L	DRA0	R/W	–	√	√	00H
FFFB3H	DMA RAM address register 0H	DRA0H		R/W	–	√		00H
FFFB4H	DMA RAM address register 1L	DRA1L	DRA1	R/W	–	√	√	00H
FFFB5H	DMA RAM address register 1H	DRA1H		R/W	–	√		00H
FFFB6H	DMA byte count register 0L	DBC0L	DBC0	R/W	–	√	√	00H
FFFB7H	DMA byte count register 0H	DBC0H		R/W	–	√		00H
FFFB8H	DMA byte count register 1L	DBC1L	DBC1	R/W	–	√	√	00H
FFFB9H	DMA byte count register 1H	DBC1H		R/W	–	√		00H
FFFB AH	DMA mode control register 0	DMC0		R/W	√	√	–	00H
FFFB BH	DMA mode control register 1	DMC1		R/W	√	√	–	00H
FFFB CH	DMA operation control register 0	DRC0		R/W	√	√	–	00H
FFFB DH	DMA operation control register 1	DRC1		R/W	√	√	–	00H
FFFB EH	Back ground event control register	BECTL		R/W	√	√	–	00H
FFFC0H	–	PFCMD <sup>Note</sup>		–	–	–	–	Undefined
FFFC2H	–	PFS <sup>Note</sup>		–	–	–	–	Undefined
FFFC4H	–	FLPMC <sup>Note</sup>		–	–	–	–	Undefined
FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD1H	Interrupt request flag register 2H	IF2H		R/W	√	√		00H
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	Interrupt mask flag register 2H	MK2H		R/W	√	√		FFH
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	Priority specification flag register 02H	PR02H		R/W	√	√		FFH
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
FFDDH	Priority specification flag register 12H	PR12H		R/W	√	√		FFH
FFFE0H	Interrupt request flag register 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	Interrupt request flag register 0H	IF0H		R/W	√	√		00H
FFFE2H	Interrupt request flag register 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	Interrupt request flag register 1H	IF1H		R/W	√	√		00H
FFFE4H	Interrupt mask flag register 0L	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H	Interrupt mask flag register 0H	MK0H		R/W	√	√		FFH
FFFE6H	Interrupt mask flag register 1L	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H	Interrupt mask flag register 1H	MK1H		R/W	√	√		FFH
FFFE8H	Priority specification flag register 00L	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H	Priority specification flag register 00H	PR00H		R/W	√	√		FFH
FFFE AH	Priority specification flag register 01L	PR01L	PR01	R/W	√	√	√	FFH
FFFE BH	Priority specification flag register 01H	PR01H		R/W	√	√		FFH
FFFE CH	Priority specification flag register 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFE DH	Priority specification flag register 10H	PR10H		R/W	√	√		FFH
FFFE EH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFE FH	Priority specification flag register 11H	PR11H		R/W	√	√		FFH

**Note** Do not directly operate this SFR, because it is to be used in the self programming library.

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
FFFF0H	Multiplication input data register A	MULA	R/W	–	–	√	0000H
FFFF1H							
FFFF2H	Multiplication input data register B	MULB	R/W	–	–	√	0000H
FFFF3H							
FFFF4H	Higher multiplication result storage register	MULOH	R	–	–	√	0000H
FFFF5H							
FFFF6H	Lower multiplication result storage register	MULOL	R	–	–	√	0000H
FFFF7H							
FFFFEH	Processor mode control register	PMC	R/W	√	√	–	00H
FFFFFH	Memory extension mode control register	MEM	R/W	√	√	–	00H

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

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

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

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

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

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation  
Describe 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.
- 16-bit manipulation  
Describe the symbol reserved by the assembler for the 16-bit manipulation instruction operand (!addr16). When specifying an address, describe an even address.

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

- Symbol  
Symbol indicating the address of an extended SFR. It is a reserved word in the RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R. When using the RA78K0R, ID78K0R-QB, and SM+ for 78K0R, 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  
“ $\surd$ ” indicates the manipulable bit unit (1, 8, or 16). “–” indicates a bit unit for which manipulation is not possible.
- After reset  
Indicates each register status upon reset signal generation.

**Caution** Do not access addresses to which 2nd SFRs are not assigned.

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

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset	
				1-bit	8-bit	16-bit		
F0017H	A/D port configuration register	ADPC	R/W	–	√	–	10H	
F0030H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H	
F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H	
F0033H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H	
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	–	00H	
F0035H	Pull-up resistor option register 5	PU5	R/W	√	√	–	00H	
F0036H	Pull-up resistor option register 6	PU6	R/W	√	√	–	00H	
F0037H	Pull-up resistor option register 7	PU7	R/W	√	√	–	00H	
F0038H	Pull-up resistor option register 8	PU8	R/W	√	√	–	00H	
F0039H	Pull-up resistor option register 9	PU9	R/W	√	√	–	00H	
F003BH	Pull-up resistor option register 11	PU11	R/W	√	√	–	00H	
F003CH	Pull-up resistor option register 12	PU12	R/W	√	√	–	00H	
F003DH	Pull-up resistor option register 13	PU13	R/W	√	√	–	00H	
F003EH	Pull-up resistor option register 14	PU14	R/W	√	√	–	00H	
F0040H	Port input mode register 0	PIM0	R/W	√	√	–	00H	
F0044H	Port input mode register 4	PIM4	R/W	√	√	–	00H	
F0049H	Port input mode register 9	PIM9	R/W	√	√	–	00H	
F004CH	Port input mode register 12	PIM12	R/W	√	√	–	00H	
F004EH	Port input mode register 14	PIM14	R/W	√	√	–	00H	
F0050H	Port output mode register 0	POM0	R/W	√	√	–	00H	
F0054H	Port output mode register 4	POM4	R/W	√	√	–	00H	
F0059H	Port output mode register 9	POM9	R/W	√	√	–	00H	
F005CH	Port output mode register 12	POM12	R/W	√	√	–	00H	
F005EH	Port output mode register 14	POM14	R/W	√	√	–	00H	
F0060H	Noise filter enable register 0	NFEN0	R/W	√	√	–	00H	
F0061H	Noise filter enable register 1	NFEN1	R/W	√	√	–	00H	
F0062H	Noise filter enable register 2	NFEN2	R/W	√	√	–	00H	
F0064H	Pull-up resistor option register 16	PU16	R/W	√	√	–	00H	
F00F0H	Peripheral enable register 0	PER0	R/W	√	√	–	00H	
F00F1H	Peripheral enable register 1	PER1	R/W	√	√	–	00H	
F00F2H	Internal high-speed oscillator trimming register	HIOTRM	R/W	–	√	–	10H	
F00F3H	Operation speed mode control register	OSMC	R/W	–	√	–	00H	
F00F4H	Regulator mode control register	RMC	R/W	–	√	–	00H	
F00FEH	BCD adjust result register	BCDADJ	R	–	√	–	Undefined	
F0100H	Serial status register 00	SSR00L	SSR00	R	–	√	√	0000H
F0101H		–			–			
F0102H	Serial status register 01	SSR01L	SSR01	R	–	√	√	0000H
F0103H		–			–			
F0104H	Serial status register 02	SSR02L	SSR02	R	–	√	√	0000H
F0105H		–			–			
F0106H	Serial status register 03	SSR03L	SSR03	R	–	√	√	0000H
F0107H		–			–			
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	–	√	√	0000H
F0109H		–			–			
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	–	√	√	0000H
F010BH		–			–			

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	–	√	√	0000H
F010DH		–			–			
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	–	√	√	0000H
F010FH		–			–			
F0110H	Serial mode register 00	SMR00		R/W	–	–	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	–	–	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	–	–	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	–	–	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	–	–	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	–	–	√	0087H
F011FH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		–			–			
F0122H	Serial channel start register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		–			–			
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		–			–			
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	–	√	√	0000H
F0127H		–			–			
F0128H	Serial output register 0	SO0		R/W	–	–	√	0F0FH
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		–			–			
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	–	√	√	0000H
F0135H		–			–			
F0140H	Serial status register 10	SSR10L	SSR10	R	–	√	√	0000H
F0141H		–			–			
F0142H	Serial status register 11	SSR11L	SSR11	R	–	√	√	0000H
F0143H		–			–			
F0144H	Serial status register 12	SSR12L	SSR12	R	–	√	√	0000H
F0145H		–			–			
F0146H	Serial status register 13	SSR13L	SSR13	R	–	√	√	0000H
F0147H		–			–			
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	–	√	√	0000H
F0149H		–			–			
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	–	√	√	0000H
F014BH		–			–			



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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F014CH	Serial flag clear trigger register 12	SIR12L	SIR12	R/W	–	√	√	0000H
F014DH		–			–			
F014EH	Serial flag clear trigger register 13	SIR13L	SIR13	R/W	–	√	√	0000H
F014FH		–			–			
F0150H	Serial mode register 10	SMR10		R/W	–	–	√	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	–	–	√	0020H
F0153H								
F0154H	Serial mode register 12	SMR12		R/W	–	–	√	0020H
F0155H								
F0156H	Serial mode register 13	SMR13		R/W	–	–	√	0020H
F0157H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	–	–	√	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	–	–	√	0087H
F015BH								
F015CH	Serial communication operation setting register 12	SCR12		R/W	–	–	√	0087H
F015DH								
F015EH	Serial communication operation setting register 13	SCR13		R/W	–	–	√	0087H
F015FH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	√	√	√	0000H
F0161H		–			–			
F0162H	Serial channel start register 1	SS1L	SS1	R/W	√	√	√	0000H
F0163H		–			–			
F0164H	Serial channel stop register 1	ST1L	ST1	R/W	√	√	√	0000H
F0165H		–			–			
F0166H	Serial clock select register 1	SPS1L	SPS1	R/W	–	√	√	0000H
F0167H		–			–			
F0168H	Serial output register 1	SO1		R/W	–	–	√	0F0FH
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1	R/W	√	√	√	0000H
F016BH		–			–			
F0174H	Serial output level register 1	SOL1L	SOL1	R/W	–	√	√	0000H
F0175H		–			–			
F0180H	Timer counter register 00	TCR00		R	–	–	√	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	–	–	√	FFFFH
F0183H								
F0184H	Timer counter register 02	TCR02		R	–	–	√	FFFFH
F0185H								
F0186H	Timer counter register 03	TCR03		R	–	–	√	FFFFH
F0187H								
F0188H	Timer counter register 04	TCR04		R	–	–	√	FFFFH
F0189H								

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F018AH	Timer counter register 05	TCR05		R	–	–	√	FFFFH
F018BH								
F018CH	Timer counter register 06	TCR06		R	–	–	√	FFFFH
F018DH								
F018EH	Timer counter register 07	TCR07		R	–	–	√	FFFFH
F018FH								
F0190H	Timer mode register 00	TMR00		R/W	–	–	√	0000H
F0191H								
F0192H	Timer mode register 01	TMR01		R/W	–	–	√	0000H
F0193H								
F0194H	Timer mode register 02	TMR02		R/W	–	–	√	0000H
F0195H								
F0196H	Timer mode register 03	TMR03		R/W	–	–	√	0000H
F0197H								
F0198H	Timer mode register 04	TMR04		R/W	–	–	√	0000H
F0199H								
F019AH	Timer mode register 05	TMR05		R/W	–	–	√	0000H
F019BH								
F019CH	Timer mode register 06	TMR06		R/W	–	–	√	0000H
F019DH								
F019EH	Timer mode register 07	TMR07		R/W	–	–	√	0000H
F019FH								
F01A0H	Timer status register 00	TSR00L	TSR00	R	–	√	√	0000H
F01A1H		–			–			
F01A2H	Timer status register 01	TSR01L	TSR01	R	–	√	√	0000H
F01A3H		–			–			
F01A4H	Timer status register 02	TSR02L	TSR02	R	–	√	√	0000H
F01A5H		–			–			
F01A6H	Timer status register 03	TSR03L	TSR03	R	–	√	√	0000H
F01A7H		–			–			
F01A8H	Timer status register 04	TSR04L	TSR04	R	–	√	√	0000H
F01A9H		–			–			
F01AAH	Timer status register 05	TSR05L	TSR05	R	–	√	√	0000H
F01ABH		–			–			
F01ACH	Timer status register 06	TSR06L	TSR06	R	–	√	√	0000H
F01ADH		–			–			
F01AEH	Timer status register 07	TSR07L	TSR07	R	–	√	√	0000H
F01AFH		–			–			
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H
F01B1H		–			–			
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	√	√	√	0000H
F01B3H		–			–			
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		–			–			

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01B6H	Timer clock select register 0	TPS0L	TPS0	R/W	–	√	√	0000H
F01B7H		–			–			
F01B8H	Timer output register 0	TO0L	TO0	R/W	–	√	√	0000H
F01B9H		–			–			
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BBH		–			–			
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	–	√	√	0000H
F01BDH		–			–			
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	–	√	√	0000H
F01BFH		–			–			
F01C0H	Timer counter register 10	TCR10		R/W	–	–	√	0000H
F01C1H								
F01C2H	Timer counter register 11	TCR11		R/W	–	–	√	0000H
F01C3H								
F01C4H	Timer counter register 12	TCR12		R/W	–	–	√	0000H
F01C5H								
F01C6H	Timer counter register 13	TCR13		R/W	–	–	√	0000H
F01C7H								
F01C8H	Timer mode register 10	TMR10		R/W	–	–	√	0000H
F01C9H								
F01CAH	Timer mode register 11	TMR11		R/W	–	–	√	0000H
F01CBH								
F01CCH	Timer mode register 12	TMR12		R/W	–	–	√	0000H
F01CDH								
F01CEH	Timer mode register 13	TMR13		R/W	–	–	√	0000H
F01CFH								
F01D0H	Timer status register 10	TSR10L	TSR10	R	–	√	√	0000H
F01D1H		–			–			
F01D2H	Timer status register 11	TSR11L	TSR11	R	–	√	√	0000H
F01D3H		–			–			
F01D4H	Timer status register 12	TSR12L	TSR12	R	–	√	√	0000H
F01D5H		–			–			
F01D6H	Timer status register 13	TSR13L	TSR13	R	–	√	√	0000H
F01D7H		–			–			
F01D8H	Timer channel enable status register 1	TE1L	TE1	R	√	√	√	0000H
F01D9H		–			–			
F01DAH	Timer channel start register 1	TS1L	TS1	R/W	√	√	√	0000H
F01DBH		–			–			
F01DCH	Timer channel stop register 1	TT1L	TT1	R/W	√	√	√	0000H
F01DDH		–			–			
F01DEH	Timer clock select register 1	TPS1L	TPS1	R/W	–	√	√	0000H
F01DFH		–			–			
F01E0H	Timer output register 1	TO1L	TO1	R/W	–	√	√	0000H
F01E1H		–			–			

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01E2H	Timer output enable register 1	TOE1L	TOE1	R/W	√	√	√	0000H
F01E3H		–			–			
F01E4H	Timer output level register 1	TOL1L	TOL1	R/W	–	√	√	0000H
F01E5H		–			–			
F01E6H	Timer output mode register 1	TOM1L	TOM1	R/W	–	√	√	0000H
F01E7H		–			–			

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

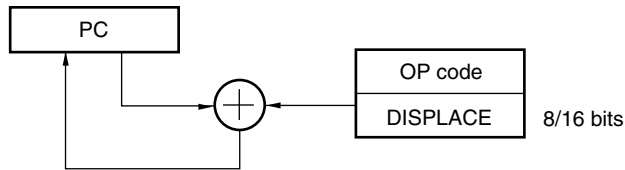
### 3.3 Instruction Address Addressing

#### 3.3.1 Relative addressing

**[Function]**

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

**Figure 3-18. Outline of Relative Addressing**



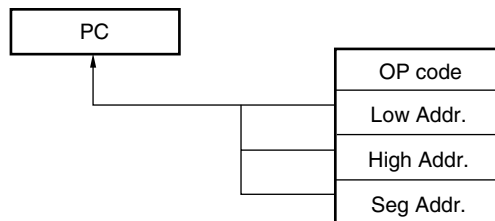
#### 3.3.2 Immediate addressing

**[Function]**

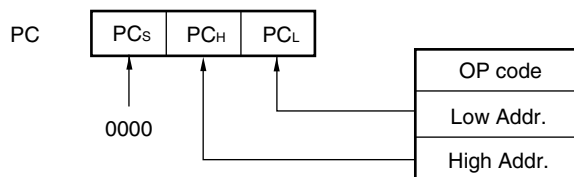
Immediate addressing stores immediate data of the instruction word in the program counter, and specifies the program address to be used as the branch destination.

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 3-19. Example of CALL !!addr20/BR !!addr20**



**Figure 3-20. Example of CALL !addr16/BR !addr16**



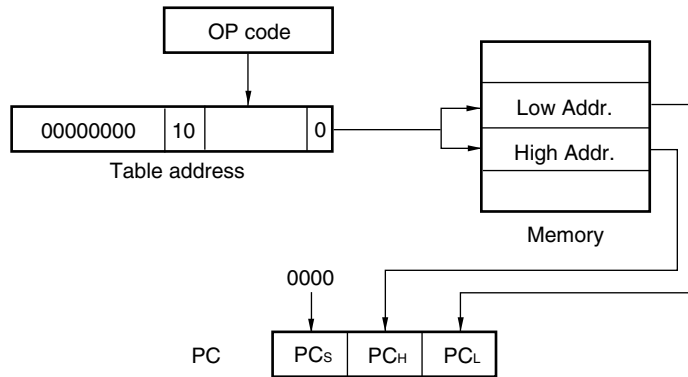
### 3.3.3 Table indirect addressing

#### [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 78K0R microcontrollers, branching is enabled only to the 64 KB space from 00000H to 0FFFFH.

**Figure 3-21. Outline of Table Indirect Addressing**

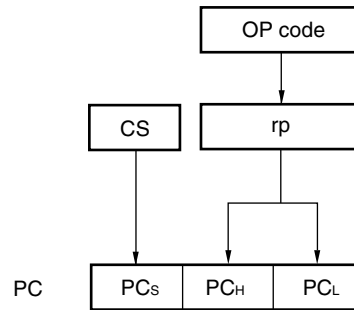


### 3.3.4 Register direct addressing

**[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 3-22. Outline of Register Direct Addressing**



### 3.4 Addressing for Processing Data Addresses

#### 3.4.1 Implied addressing

**[Function]**

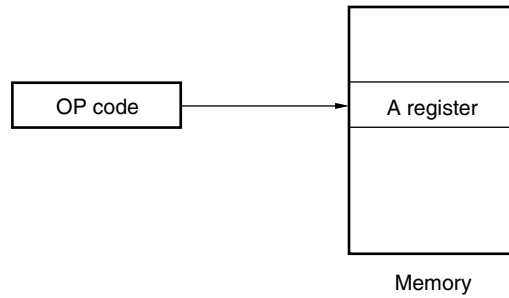
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

**[Operand format]**

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

Implied addressing can be applied only to MULU X.

**Figure 3-23. Outline of Implied Addressing**



#### 3.4.2 Register addressing

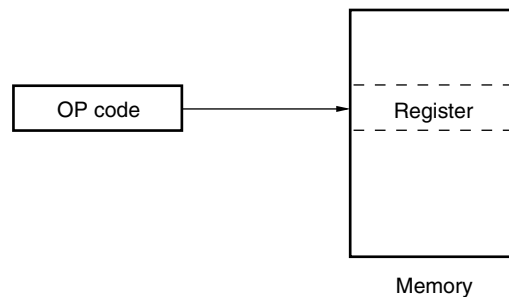
**[Function]**

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 3-24. Outline of Register Addressing**





### 3.4.3 Direct addressing

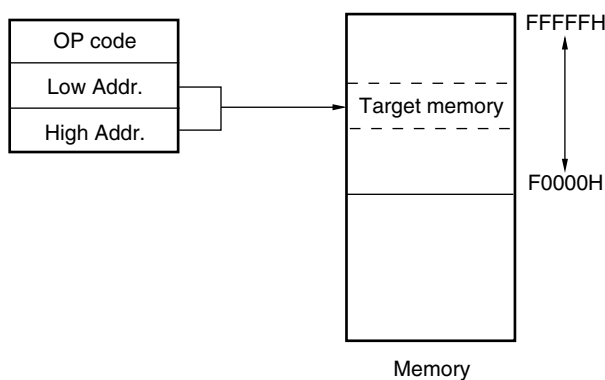
**[Function]**

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

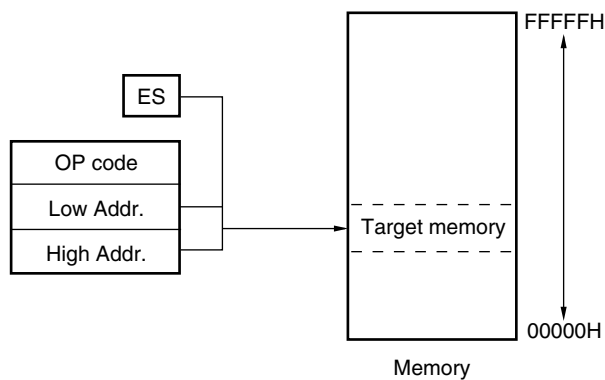
**[Operand format]**

Identifier	Description
ADDR16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES: ADDR16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

**Figure 3-25. Example of ADDR16**



**Figure 3-26. Example of ES:ADDR16**



### 3.4.4 Short direct addressing

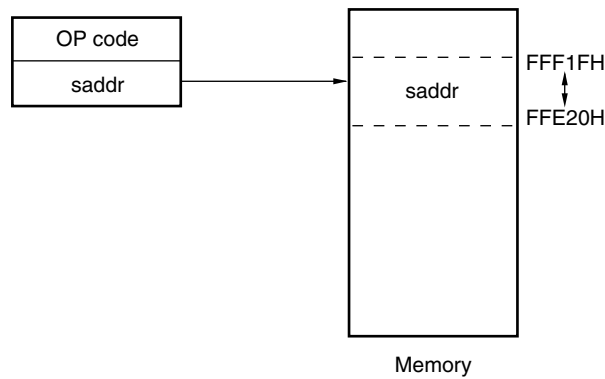
**[Function]**

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

**[Operand format]**

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

**Figure 3-27. 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 16-bit or 20-bit immediate data is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

### 3.4.5 SFR addressing

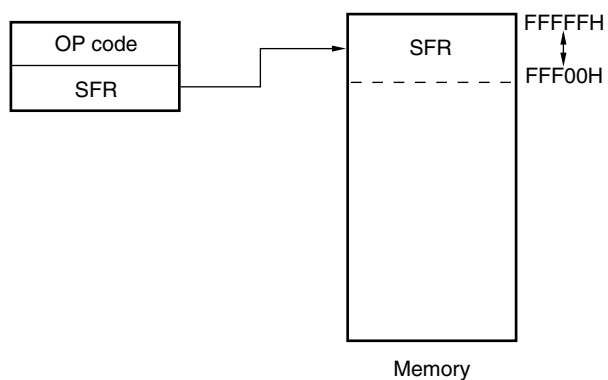
**[Function]**

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 3-28. Outline of SFR Addressing**



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

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

Figure 3-29. Example of [DE], [HL]

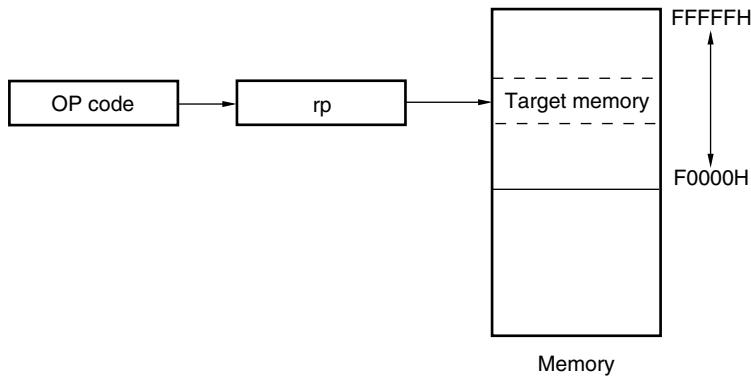
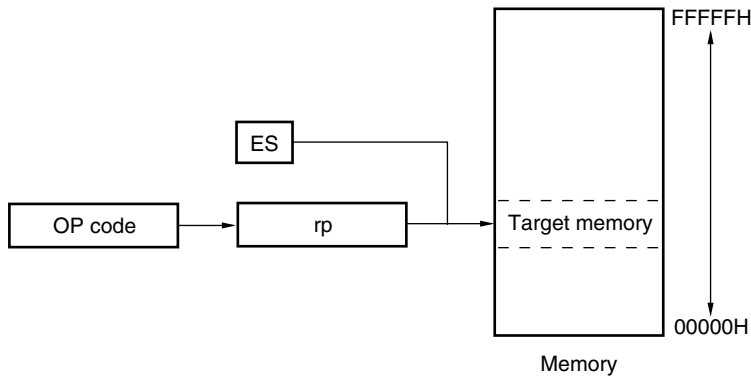


Figure 3-30. Example of ES:[DE], ES:[HL]



## 3.4.7 Based addressing

**[Function]**

Based addressing uses the contents of a register pair specified with the instruction word 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 3-31. Example of [SP+byte]

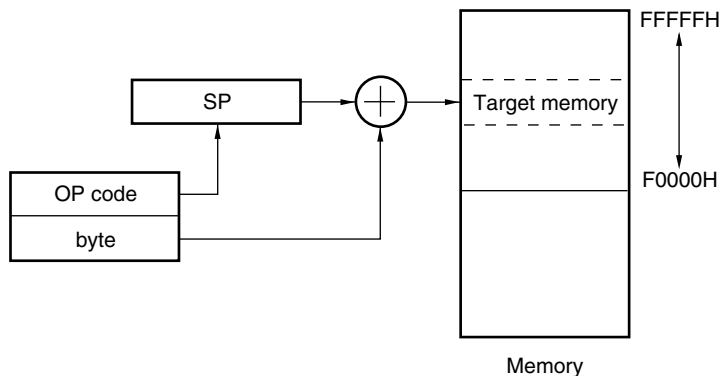


Figure 3-32. Example of [HL + byte], [DE + byte]

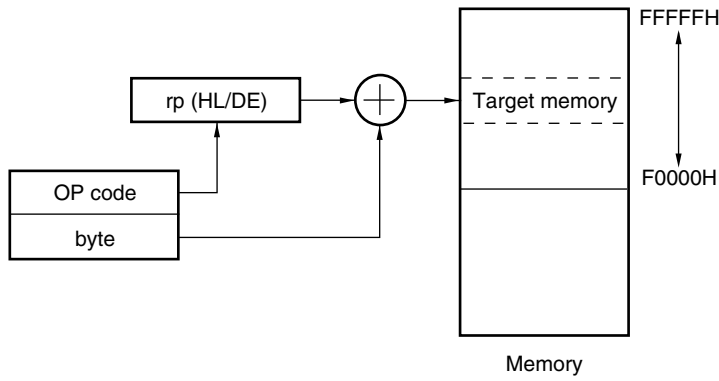


Figure 3-33. Example of word[B], word[C]

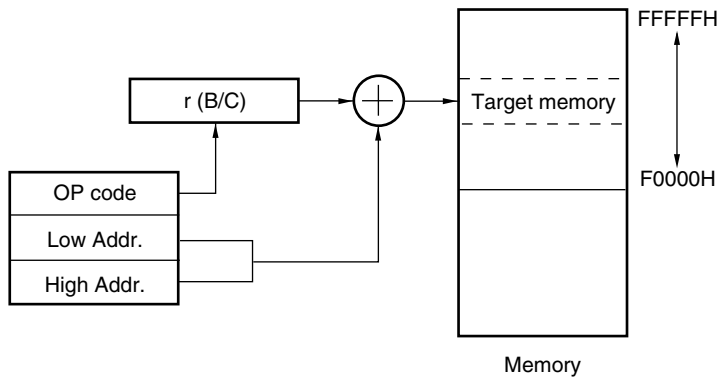


Figure 3-34. Example of word[BC]

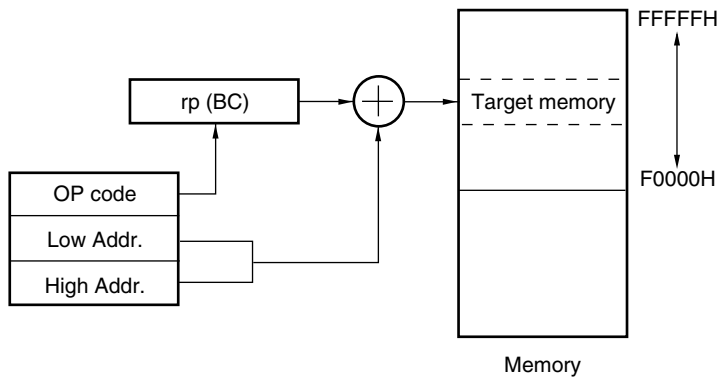


Figure 3-35. Example of ES:[HL + byte], ES:[DE + byte]

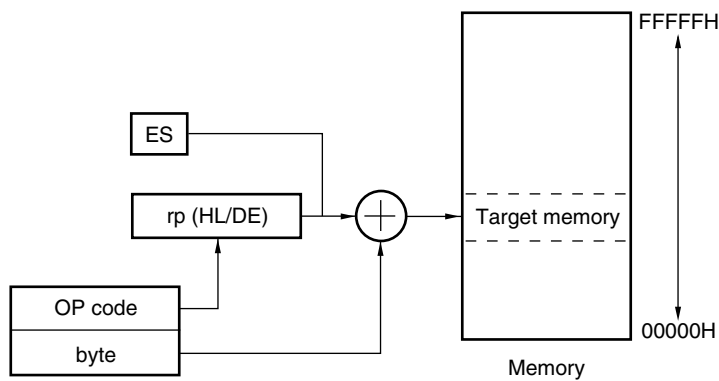


Figure 3-36. Example of ES:word[B], ES:word[C]

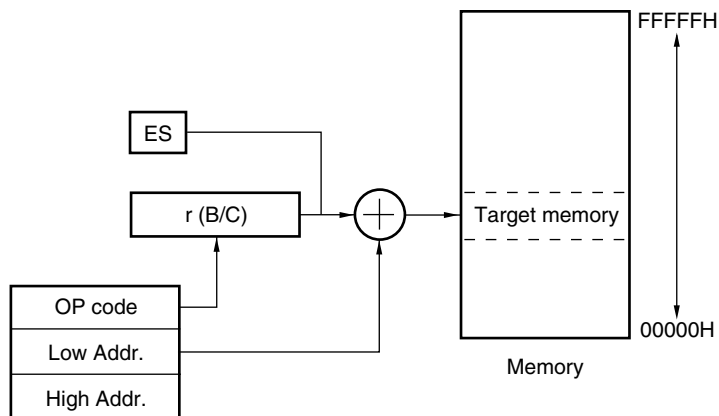
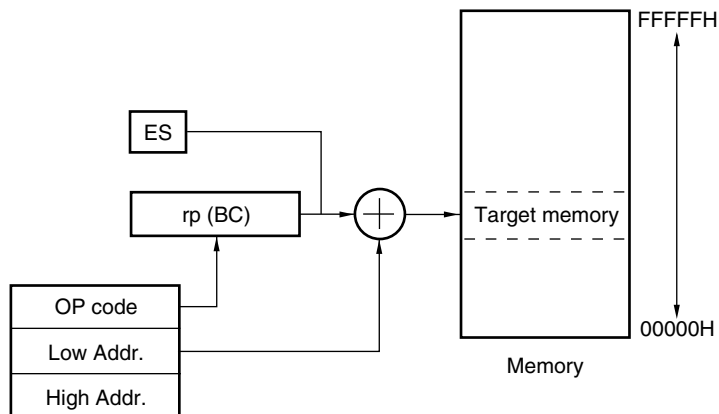


Figure 3-37. Example of ES:word[BC]



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 3-38. Example of [HL+B], [HL+C]

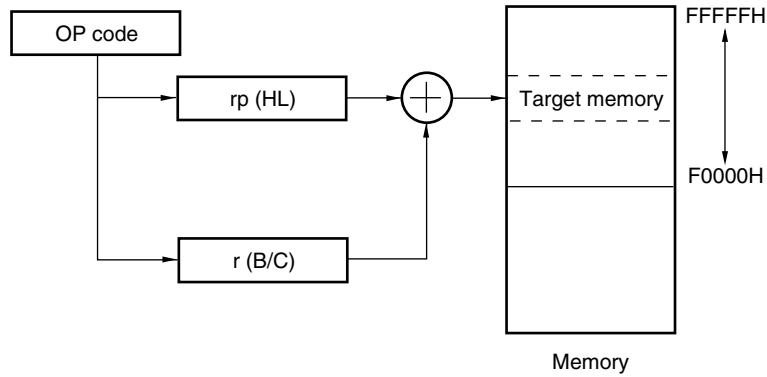
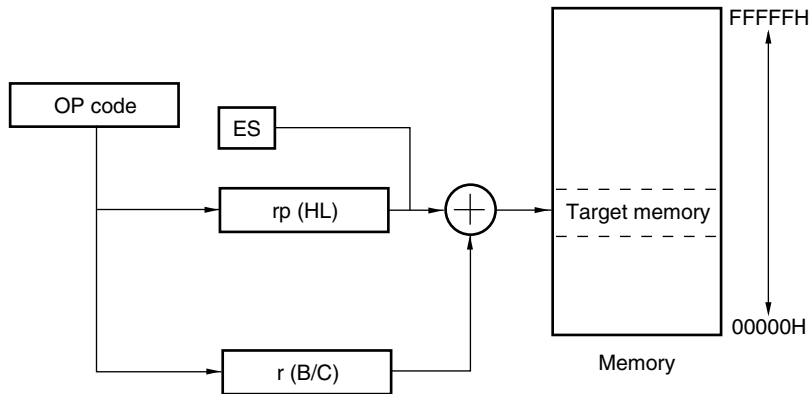


Figure 3-39. Example of ES:[HL+B], ES:[HL+C]





### 3.4.9 Stack addressing

**[Function]**

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

Stack addressing is applied only to the internal RAM area.

**[Operand format]**

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

## CHAPTER 4 PORT FUNCTIONS

### 4.1 Port Functions

There are five types of pin I/O buffer power supplies:  $AV_{REF0}$ ,  $AV_{REF1}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ , and  $V_{DD}$ . The relationship between these power supplies and the pins is shown below.

**Table 4-1. Pin I/O Buffer Power Supplies**

Power Supply	Corresponding Pins
$AV_{REF0}$	P20 to P27, P150 to P157
$AV_{REF1}$	P110, P111
$EV_{DD0}$ , $EV_{DD1}$	<ul style="list-style-type: none"> <li>• Port pins other than P20 to P27, P110, P111, P121 to P124, and P150 to P157</li> <li>• <math>\overline{RESET}</math> and FLMD0 pins</li> </ul>
$V_{DD}$	<ul style="list-style-type: none"> <li>• P121 to P124</li> <li>• Non-port pins (excluding <math>\overline{RESET}</math> and FLMD0 pins)</li> </ul>

78K0R/KH3 products are provided with the ports shown in Figure 4-1, which enable variety of control operations. The functions of each port are shown in Table 4-2.

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**.

**Figure 4-1. Port Types**

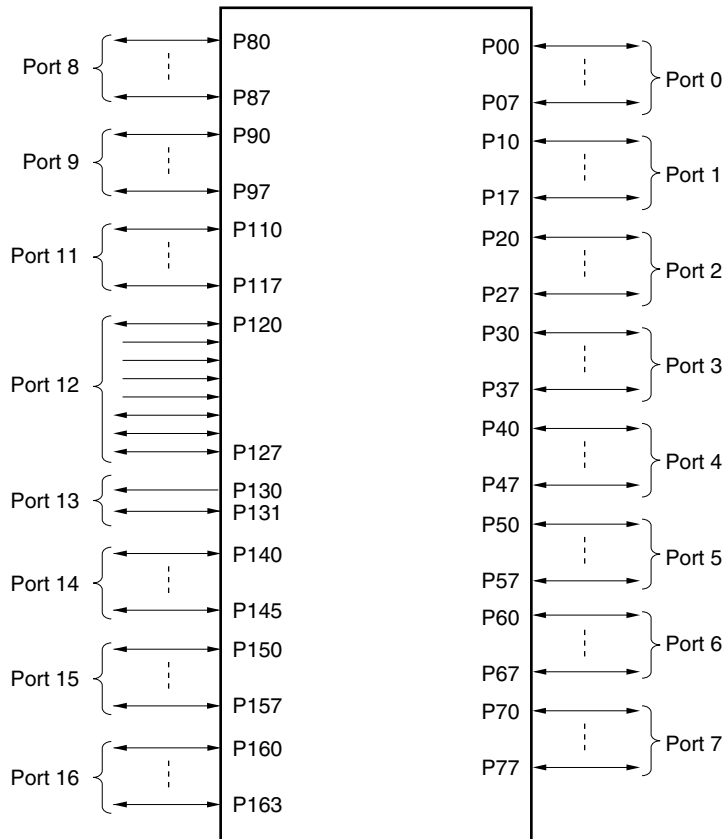


Table 4-2. Port Functions (1/3)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 8-bit I/O port. Input of P03 and P04 can be set to TTL input buffer. Output of P02 to P04 can be set to N-ch open-drain output ( $V_{DD}$ tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI00
P01				TO00
P02				SO10/TxD1
P03				SI10/RxD1/SDA10
P04				SCK10/SCL10
P05				CLKOUT
P06				WAIT
P07				–
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCK00/EX24
P11				SI00/RxD0/EX25
P12				SO00/TxD0/EX26
P13				TxD3/EX27
P14				RxD3/EX28
P15				RTCDIV/RTCCL/EX29
P16				TI01/TO01/INTP5/EX30
P17				TI02/TO02/EX31
P20 to P27	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI0 to ANI7
P30	I/O	Port 3. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	RTC1HZ/INTP3
P31				TI03/TO03/INTP4
P32 to P37				–
P40 <sup>Note</sup>	I/O	Port 4. 8-bit I/O port. Input of P43 and P44 can be set to TTL input buffer. Output of P43 and P45 can be set to N-ch open-drain output ( $V_{DD}$ tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOOL0
P41				TOOL1
P42				TI04/TO04
P43				SCK01
P44				SI01
P45				SO01
P46				INTP1/TI05/TO05
P47				INTP2
P50 to P57	I/O	Port 5. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	EX8 to EX15

**Note** If on-chip debugging is enabled by using an option byte, be sure to pull up the P40/TOOL0 pin externally (see **Caution** in 2.2.5 P40 to P47 (port 4)).

Table 4-2. Port Functions (2/3)

Function Name	I/O	Function	After Reset	Alternate Function
P60	I/O	Port 6. 8-bit I/O port. Output of P60 to P63 can be set to N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units. For only P64 to P67, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCL0
P61				SDA0
P62				–
P63				–
P64				$\overline{\text{RD}}$
P65				$\overline{\text{WR0}}$
P66				$\overline{\text{WR1}}$
P67				ASTB
P70 to P73	I/O	Port 7. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	KR0/EX16 to KR3/EX19
P74 to P77				KR4/EX20/INTP8 to KR7/EX23/INTP11
P80 to P87	I/O	Port 8. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	EX0 to EX7
P90 to P93	I/O	Port 9. 8-bit I/O port. Input of P95 and P96 can be set to TTL input buffer. Output of P95 to P97 can be set to the N-ch open-drain output ( $V_{\text{DD}}$ tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	EX32 to EX35
P94				–
P95				$\overline{\text{SCK11}}/\text{SCL11}$
P96				SI11/SDA11
P97				SO11
P110	I/O	Port 11. 8-bit I/O port. Input/output can be specified in 1-bit units. For only P112 to P117, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	ANO0
P111				ANO1
P112 to P117				–
P120	I/O	Port 12. 4-bit I/O port and 4-bit input port. Input of P125 and P126 can be set to TTL input buffer. Output of P125 to P127 can be set to the N-ch open-drain output ( $V_{\text{DD}}$ tolerance). For only P120 and P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121	Input			X1
P122				X2/EXCLK
P123				XT1
P124				XT2
P125	I/O			$\overline{\text{SCK21}}/\text{SCL21}$
P126				SI21/SDA21
P127				SO21

**Table 4-2. Port Functions (3/3)**

Function Name	I/O	Function	After Reset	Alternate Function
P130	Output	Port 13. 1-bit output port and 1-bit I/O port.	Output port	–
P131	I/O	For only P131 use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI06/TO06
P140	I/O	Port 14. 6-bit I/O port. Input of P142 and P143 can be set to TTL input buffer. Output of P142 to P144 can be set to the N-ch open-drain output ( $V_{DD}$ tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	PCLBUZ0/INTP6
P141				PCLBUZ1/INTP7
P142				SCK20/SCL20
P143				SI20/RxD2/SDA20
P144				SO20/TxD2
P145				TI07/TO07
P150 to P157	I/O	Port 15. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI8 to ANI15
P160 to P163	I/O	Port 16. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI10/TO10 to TI13/TO13

## 4.2 Port Configuration

Ports include the following hardware.

**Table 4-3. Port Configuration**

Item	Configuration
Control registers	Port mode registers (PM0 to PM9, PM11 to PM16) Port registers (P0 to P9, P11 to P16) Pull-up resistor option registers (PU0, PU1, PU3 to PU9, PU11 to PU14, PU16) Port input mode registers (PIM0, PIM4, PIM9, PIM12, PIM14) Port output mode registers (POM0, POM4, POM9, POM12, POM14) A/D port configuration register (ADPC)
Port	Total: 116 (CMOS I/O: 107, CMOS input: 4, CMOS output: 1, N-ch open drain I/O: 4)
Pull-up resistor	Total: 89

### 4.2.1 Port 0

Port 0 is an 8-bit I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P07 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Input to the P03 and P04 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 0 (PIM0).

Output from the P02 to P04 pins can be specified as N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 0 (POM0).

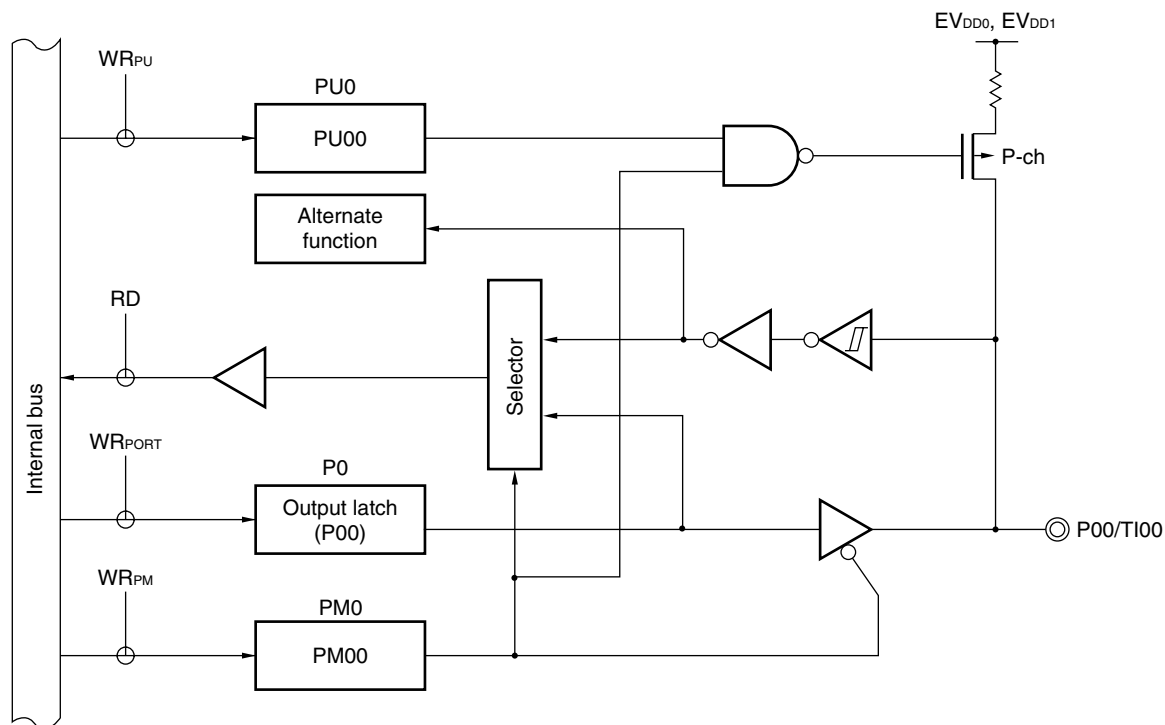
This port can also be used for timer I/O, serial interface data I/O, clock I/O, internal system clock output, and external wait signal input.

Reset signal generation sets port 0 to input mode.

Figures 4-2 to 4-8 show block diagrams of port 0.

- Cautions**
- 1. To use P01/TO00 as a general-purpose port, set bit 0 (TO00) of timer output register 0 (TO0) and bit 0 (TOE00) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.**
  - 2. To use P02/SO10/TxD1, P03/SI10/RxD1/SDA10, or P04/ $\overline{\text{SCK10}}$ /SCL10 as a general-purpose port, note the serial array unit 0 setting. For details, refer to the following tables.**
    - **Table 13-7 Relationship Between Register Settings and Pins (Channel 2 of Unit 0: CSI10, UART1 Transmission, IIC10)**
    - **Table 13-8 Relationship Between Register Settings and Pins (Channel 3 of Unit 0: CSI11, UART1 Reception, IIC11)**

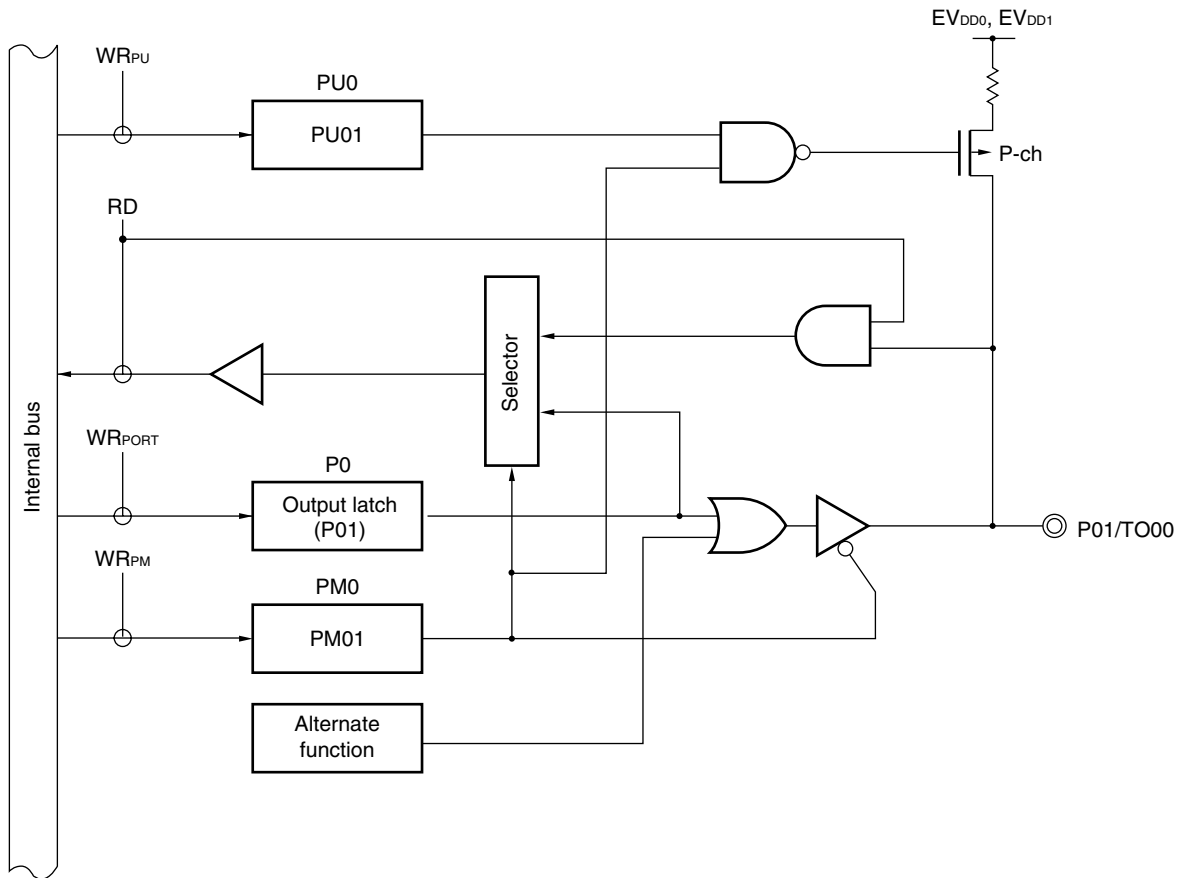
Figure 4-2. Block Diagram of P00



- P0: Port register 0
- PU0: Pull-up resistor option register 0
- PM0: Port mode register 0
- RD: Read signal
- WR<sub>xx</sub>: Write signal

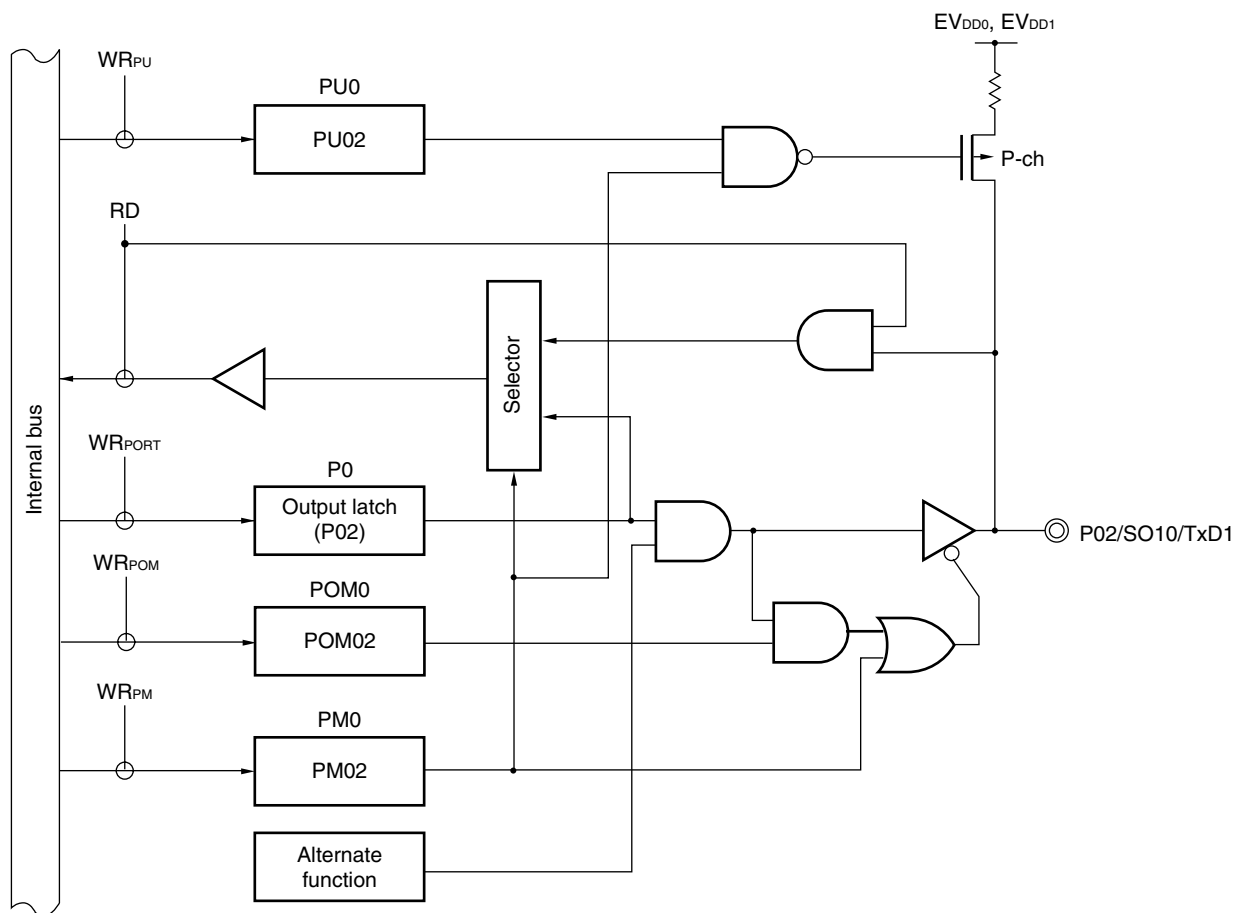


Figure 4-3. Block Diagram of P01



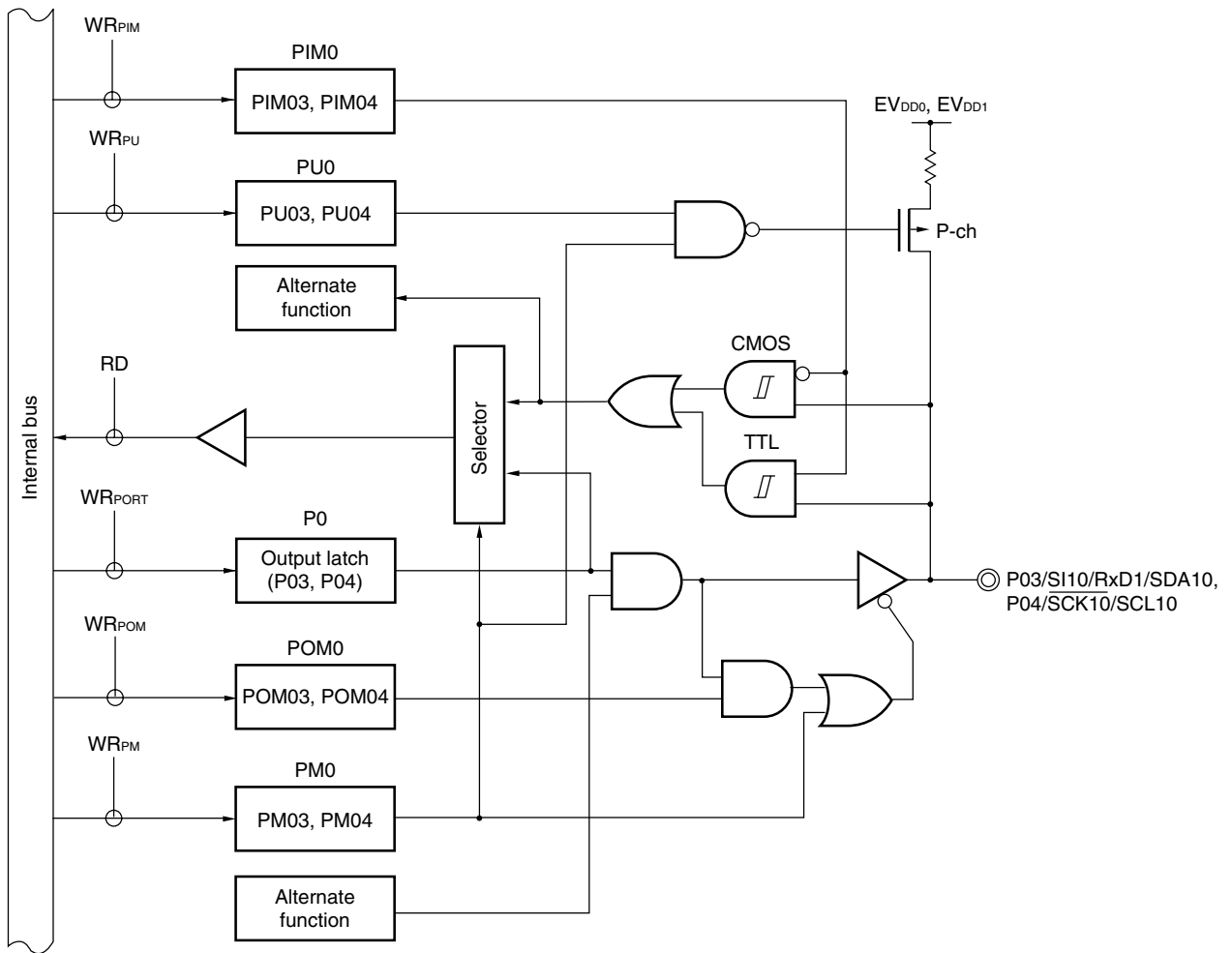
- P0: Port register 0
- PU0: Pull-up resistor option register 0
- PM0: Port mode register 0
- RD: Read signal
- $WR_{\times\times}$ : Write signal

Figure 4-4. Block Diagram of P02



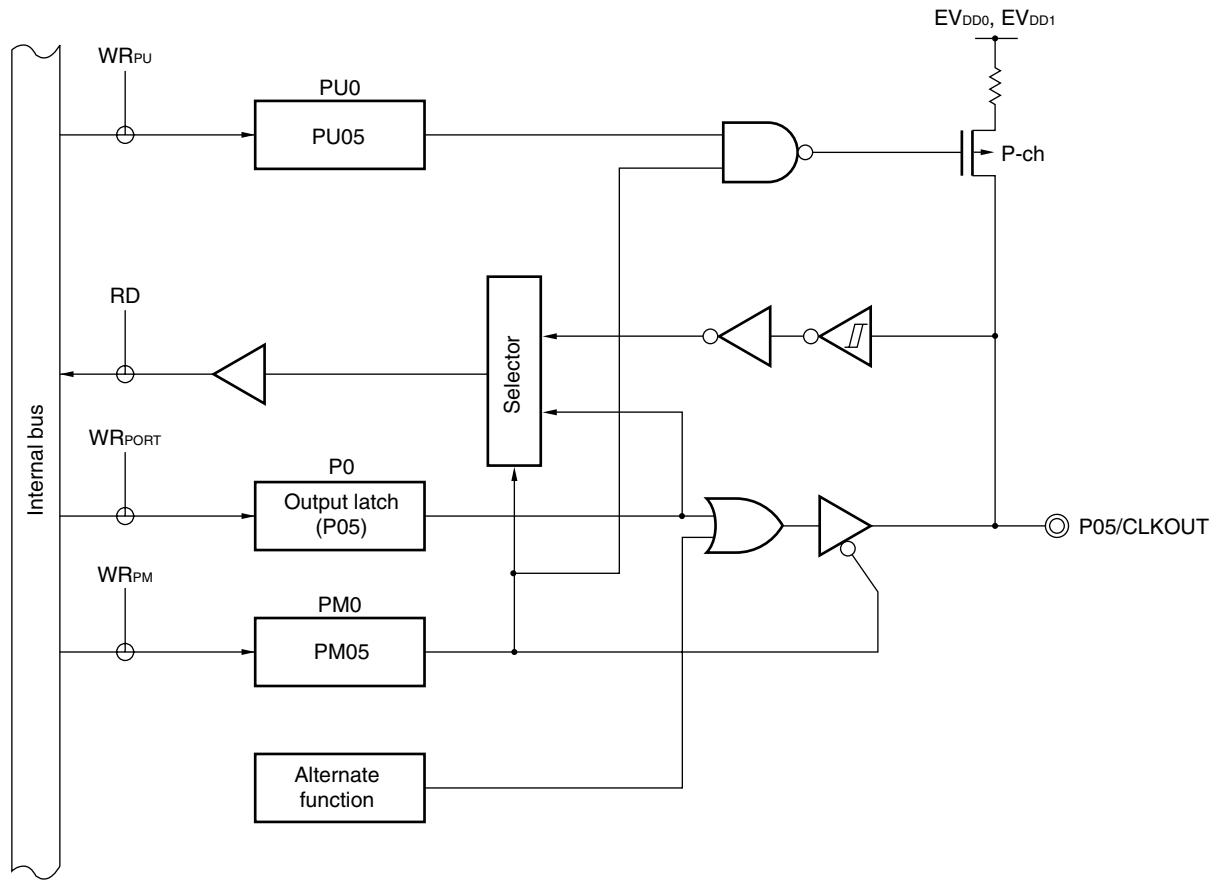
- P0: Port register 0
- PU0: Pull-up resistor option register 0
- PM0: Port mode register 0
- POM0: Port output mode register 0
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-5. Block Diagram of P03 and P04



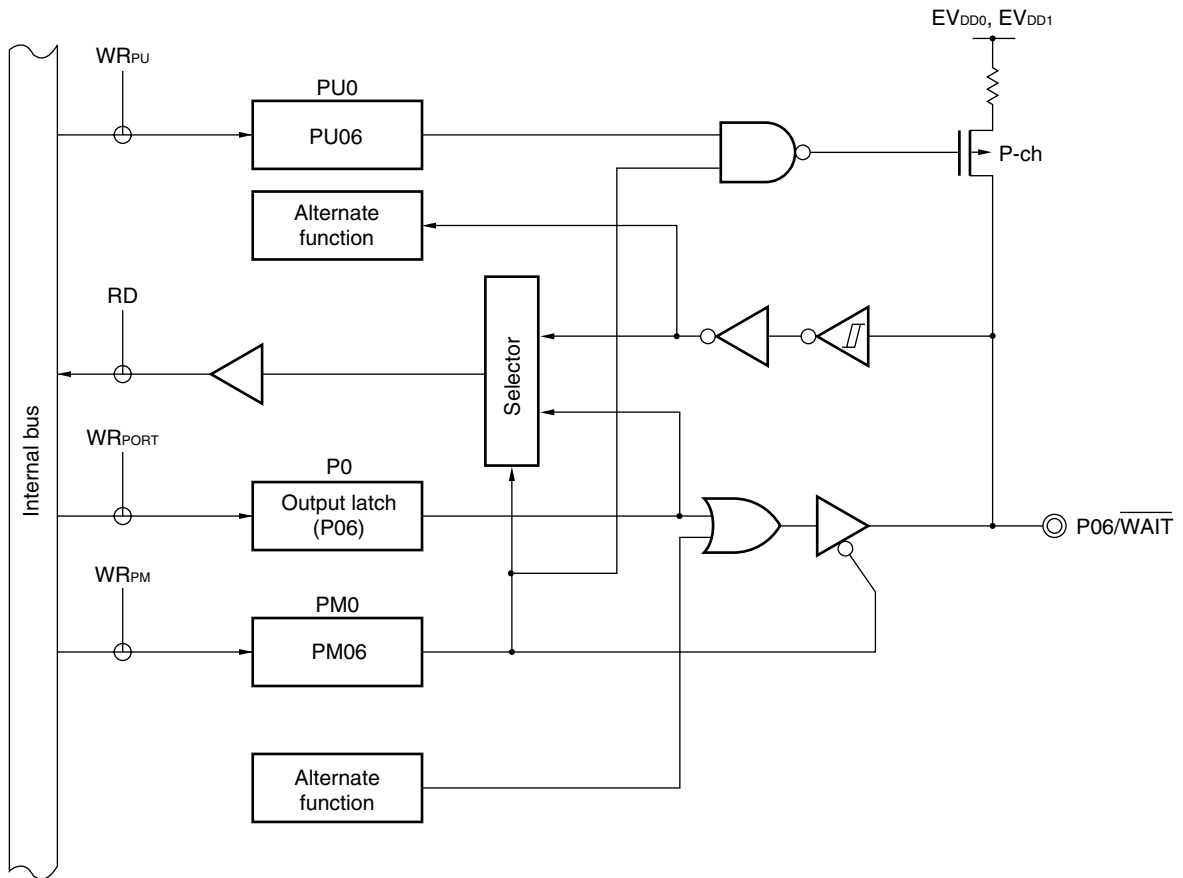
- P0: Port register 0
- PU0: Pull-up resistor option register 0
- PM0: Port mode register 0
- PIM0: Port input mode register 0
- POM0: Port output mode register 0
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-6. Block Diagram of P05



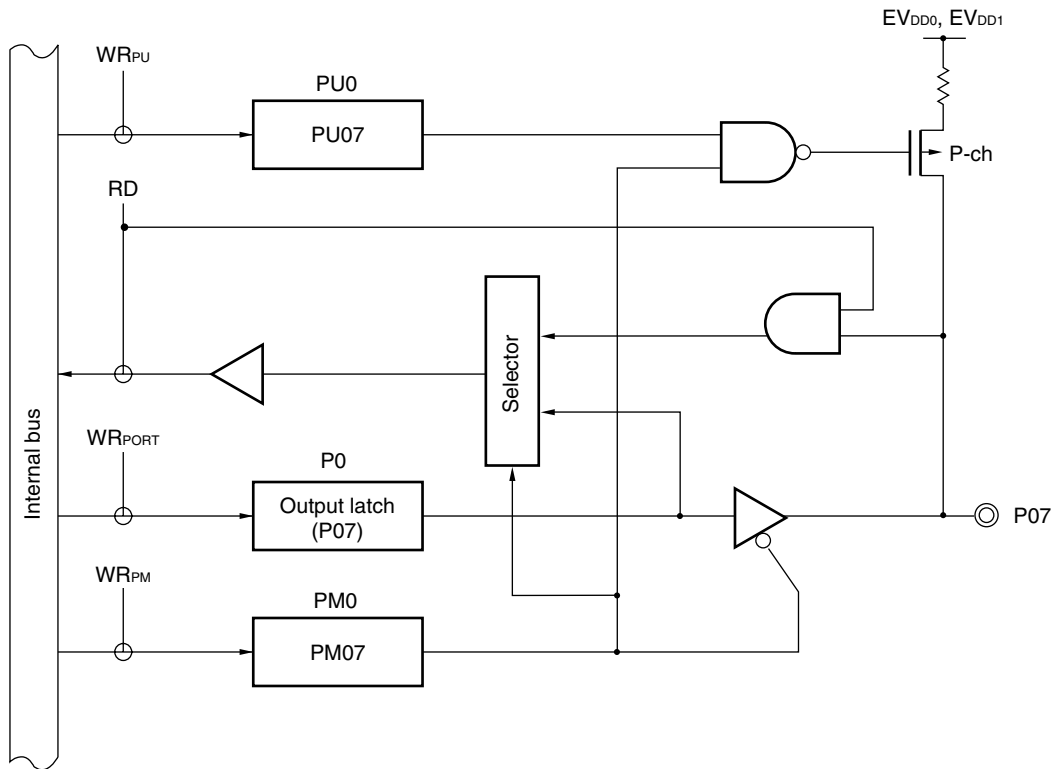
- P0: Port register 0
- PU0: Pull-up resistor option register 0
- PM0: Port mode register 0
- RD: Read signal
- $WR_{xx}$ : Write signal

Figure 4-7. Block Diagram of P06



- P0: Port register 0
- PU0: Pull-up resistor option register 0
- PM0: Port mode register 0
- RD: Read signal
- $WR_{xx}$ : Write signal

Figure 4-8. Block Diagram of P07



- P0: Port register 0
- PU0: Pull-up resistor option register 0
- PM0: Port mode register 0
- RD: Read signal
- $WR_{xx}$ : Write signal

### 4.2.2 Port 1

Port 1 is an 8-bit I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

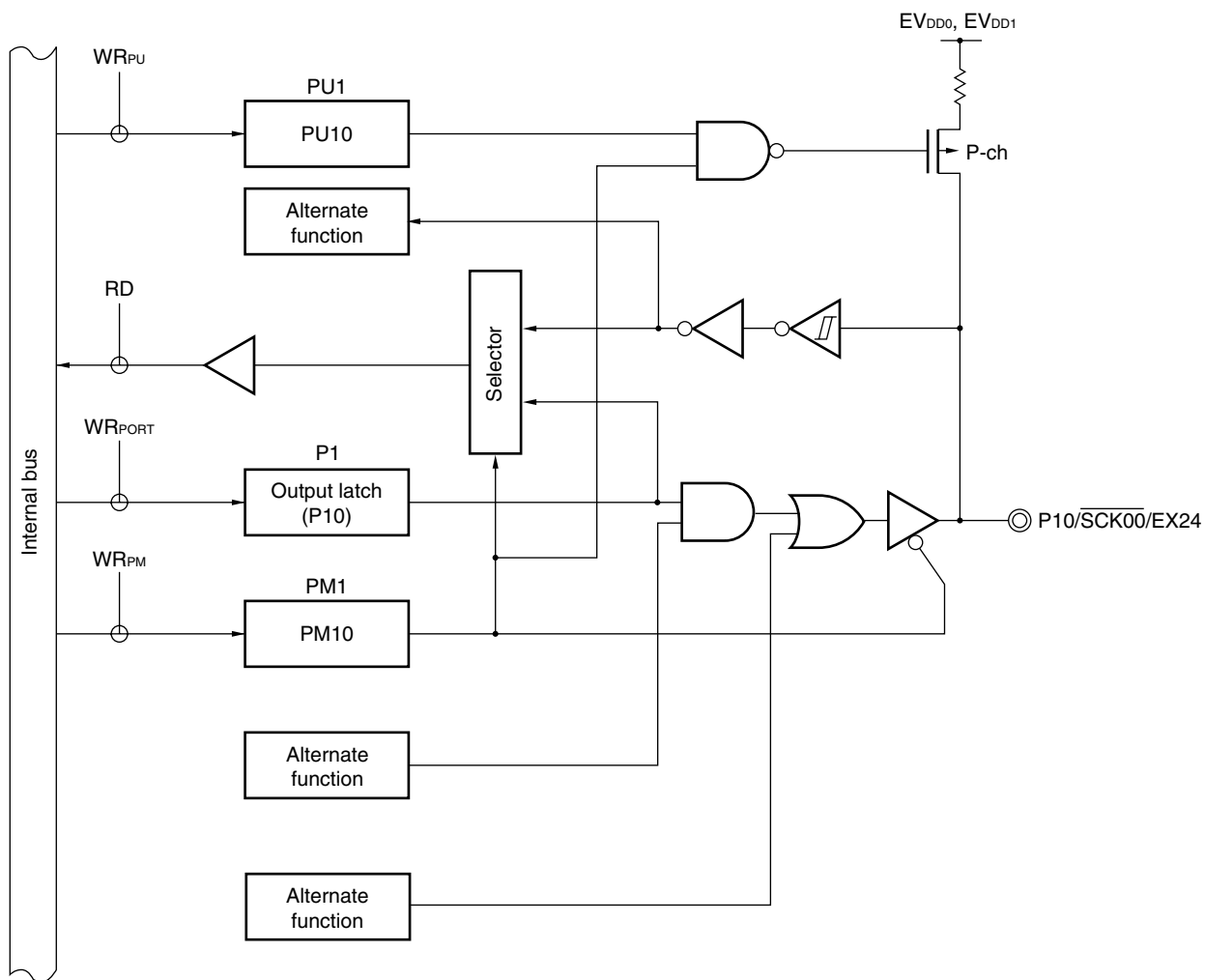
This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, timer I/O, real-time counter clock output, and external expansion output (address bus).

Reset signal generation sets port 1 to input mode.

Figures 4-9 to 4-13 show block diagrams of port 1.

- Cautions**
1. To use P10/SCK00/EX24, P11/SI00/RxD0/EX25, P12/SO00/TxD0/EX26, P13/TxD3/EX27, or P14/RxD3/EX28 as a general-purpose port, note the serial array unit setting. For details, refer to the following tables.
    - Table 13-5 Relationship Between Register Settings and Pins (Channel 0 of Unit 0: CSI00, UART0 Transmission)
    - Table 13-6 Relationship Between Register Settings and Pins (Channel 1 of Unit 0: CSI01, UART0 Reception)
    - Table 13-11 Relationship Between Register Settings and Pins (Channel 2 of Unit 1: UART3 Transmission)
    - Table 13-12 Relationship Between Register Settings and Pins (Channel 3 of Unit 1: UART3 Reception)
  2. To use P16/TI01/TO01/INTP5/EX30 or P17/TI02/TO02/EX31 as a general-purpose port, set bits 1 and 2 (TO01, TO02) of timer output register 0 (TO0) and bits 1 and 2 (TOE01, TOE02) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.
  3. To use P15/RTCDIV/RTCCL/EX29 as a general-purpose port, set bit 4 (RCLOE0) of Real-time counter control register 0 (RTCC0) and bit 6 (RCLOE2) of Real-time counter control register 2 (RTCC2) to “0”, which is the same as their default status setting.
  4. Do not enable outputting other alternate functions when the external expansion output (address bus) function is used.

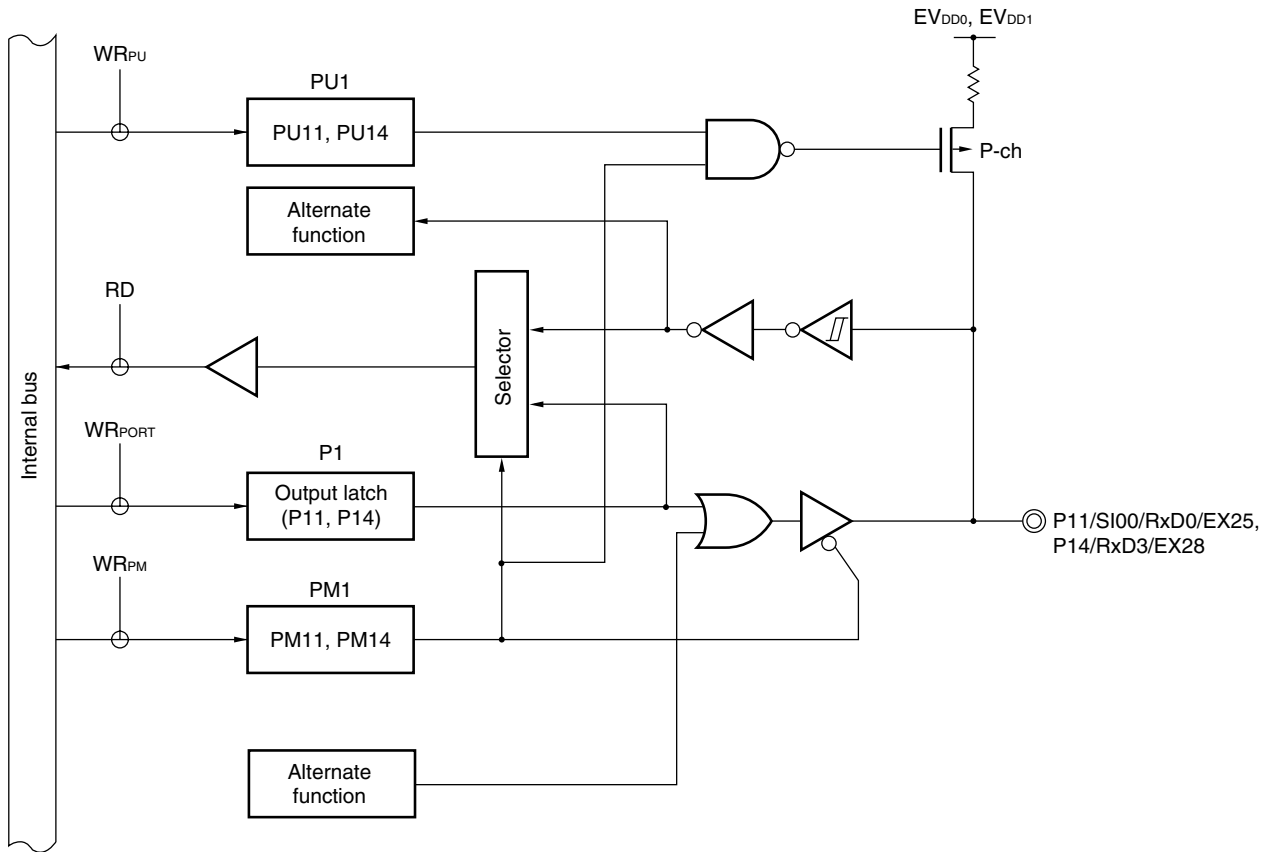
Figure 4-9. Block Diagram of P10



- P1: Port register 1
- PU1: Pull-up resistor option register 1
- PM1: Port mode register 1
- RD: Read signal
- WR<sub>xx</sub>: Write signal

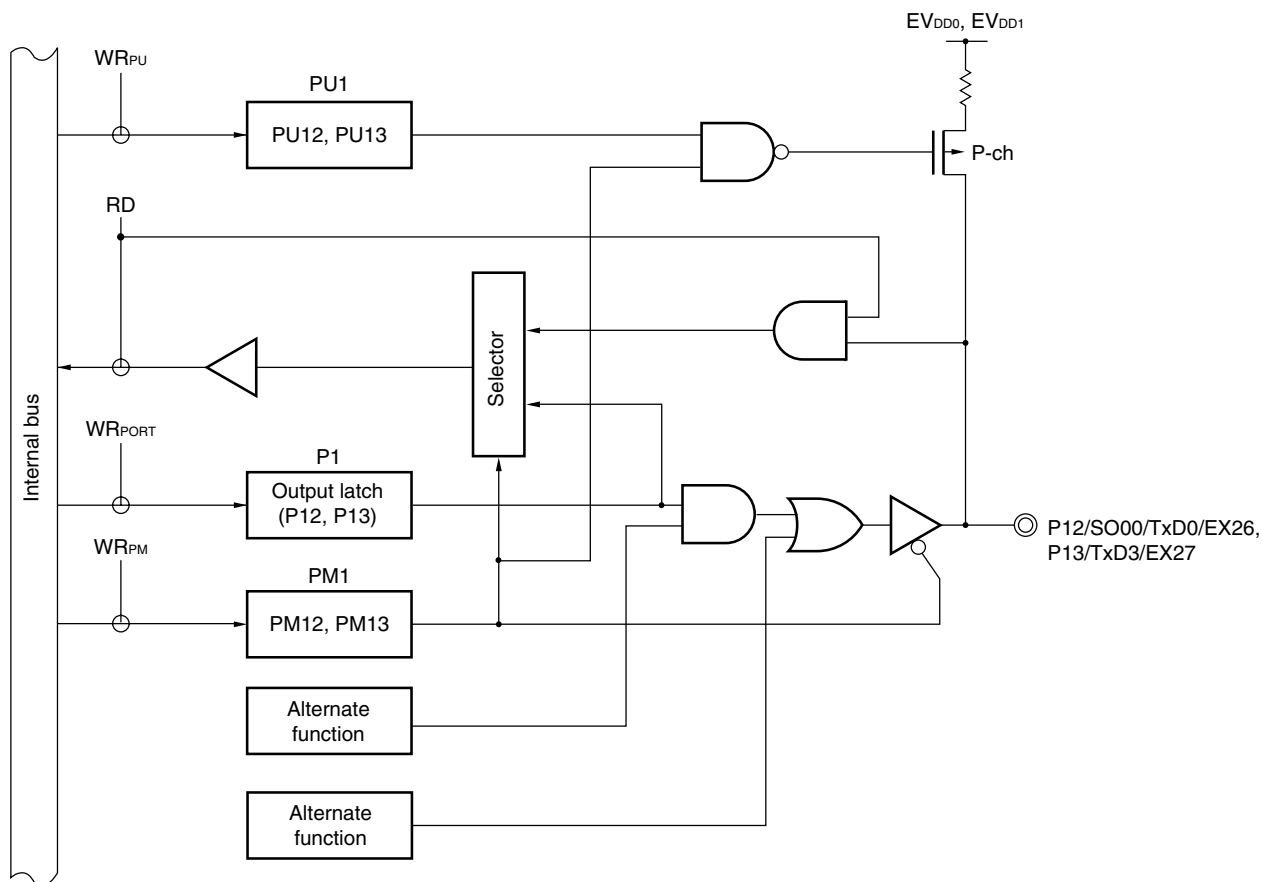


Figure 4-10. Block Diagram of P11 and P14



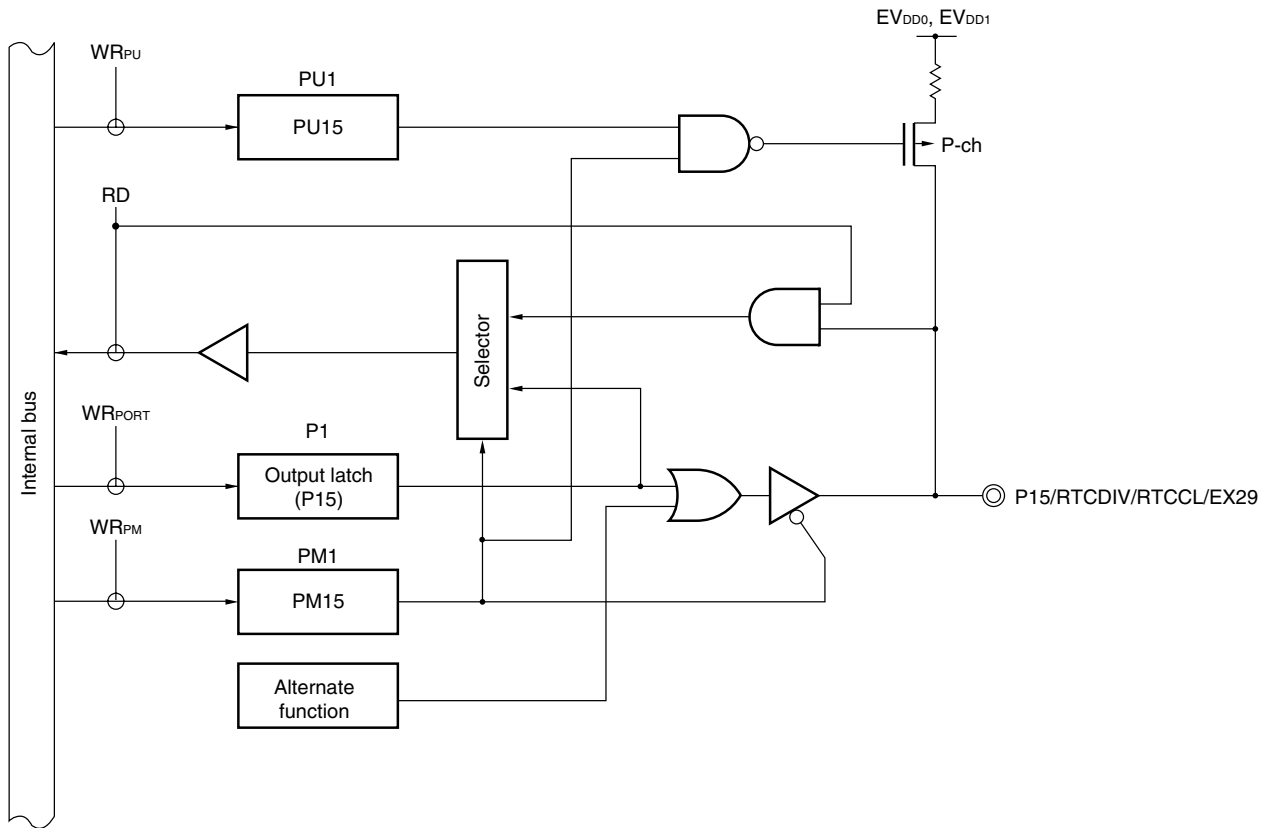
- P1: Port register 1
- PU1: Pull-up resistor option register 1
- PM1: Port mode register 1
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-11. Block Diagram of P12 and P13



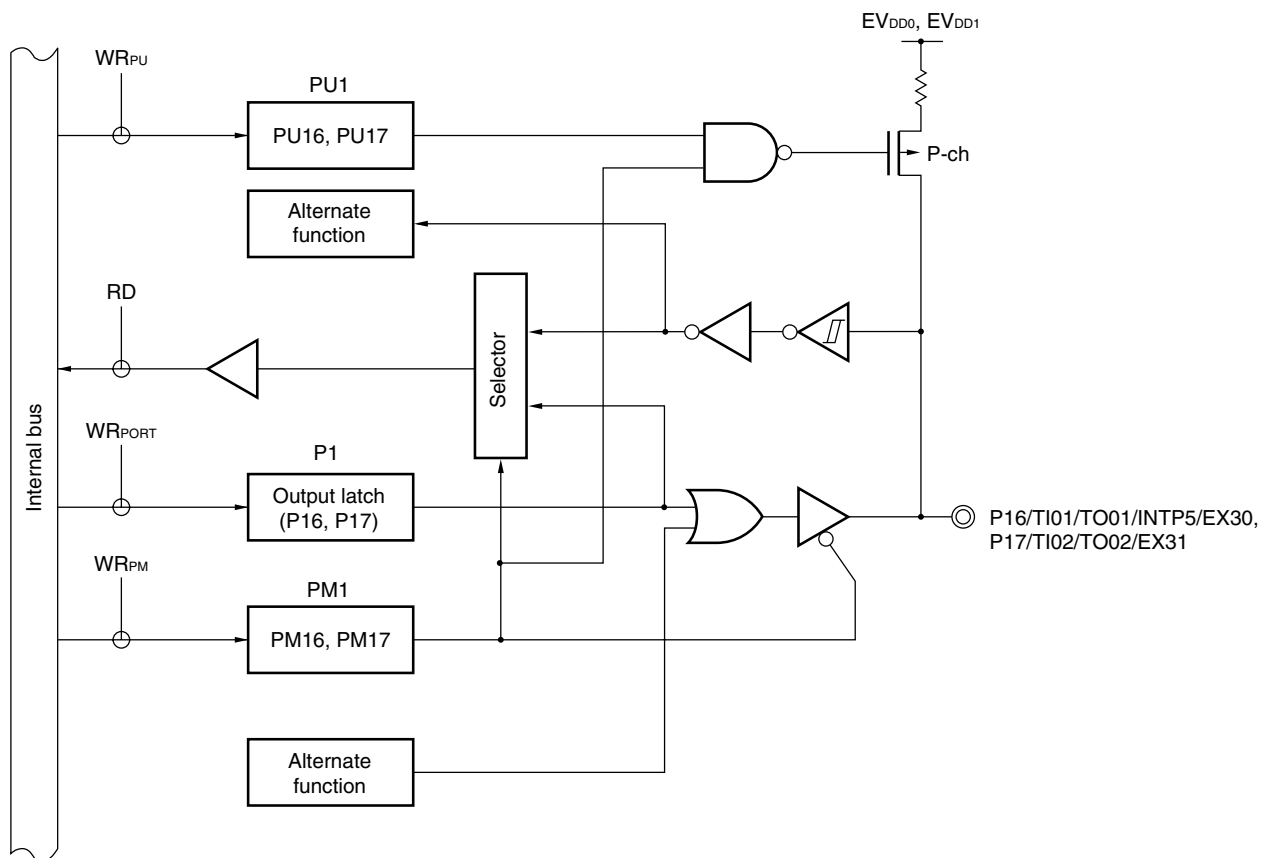
- P1: Port register 1
- PU1: Pull-up resistor option register 1
- PM1: Port mode register 1
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-12. Block Diagram of P15



- P1: Port register 1
- PU1: Pull-up resistor option register 1
- PM1: Port mode register 1
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-13. Block Diagram of P16 and P17



- P1: Port register 1
- PU1: Pull-up resistor option register 1
- PM1: Port mode register 1
- RD: Read signal
- WR<sub>xx</sub>: Write signal

**4.2.3 Port 2**

Port 2 is an 8-bit I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units using port mode register 2 (PM2).

This port can also be used for A/D converter analog input.

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

To use P20/ANI0 to P27/ANI7 as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM2.

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

**Table 4-4. Setting Functions of P20/ANI0 to P27/ANI7 Pins**

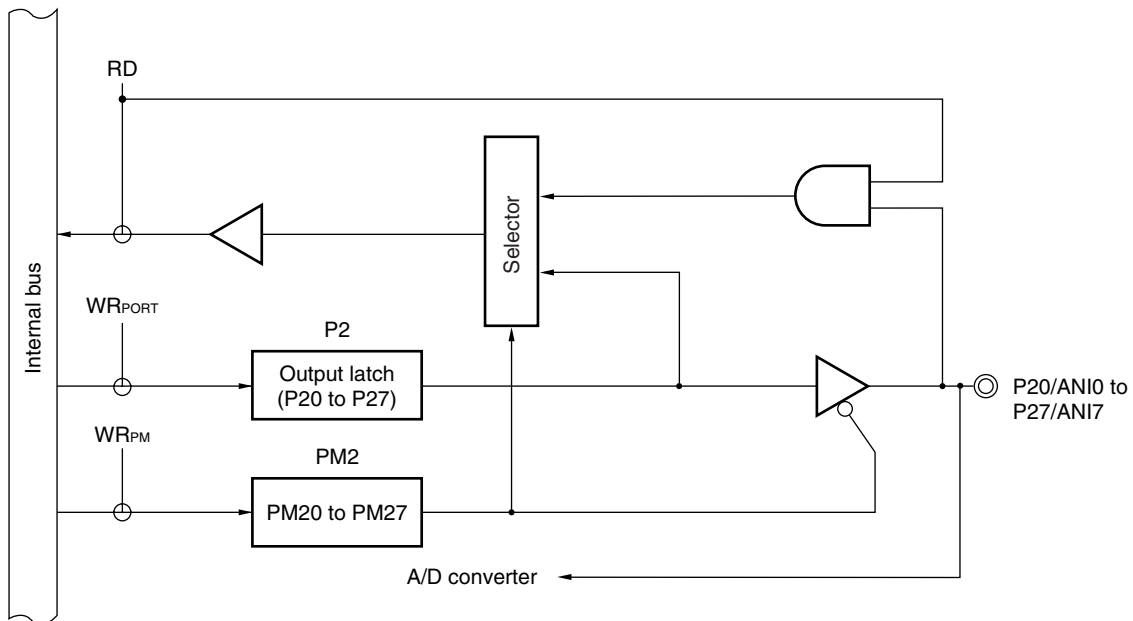
ADPC	PM2	ADS	P20/ANI0 to P27/ANI7 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
	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

All P20/ANI0 to P27/ANI7 are set in the digital input mode when the reset signal is generated.

Figure 4-14 shows a block diagram of port 2.

**Caution** See 2.2.17 AV<sub>REF0</sub> for the voltage to be applied to the AV<sub>REF0</sub> pin when using port 2 as a digital I/O.

Figure 4-14. Block Diagram of P20 to P27



- P2: Port register 2
- PM2: Port mode register 2
- RD: Read signal
- WR<sub>xx</sub>: Write signal

#### 4.2.4 Port 3

Port 3 is an 8-bit I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 to P37 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

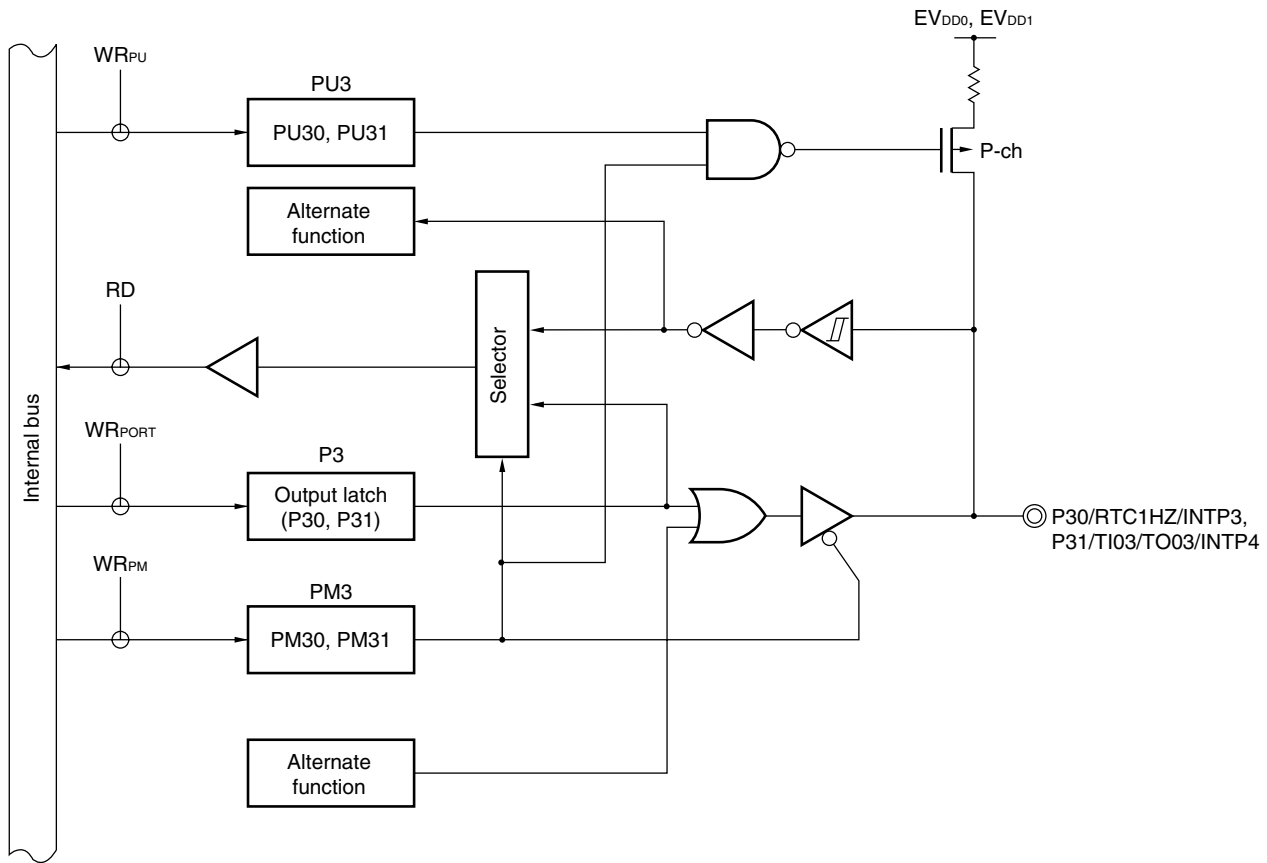
This port can also be used for external interrupt request input, timer I/O, and real-time counter correction clock output.

Reset signal generation sets port 3 to input mode.

Figures 4-15 and 4-16 show block diagrams of port 3.

- Cautions**
1. To use P31/TI03/TO03/INTP4 as a general-purpose port, set bit 3 (TO03) of timer output register 0 (TO0) and bit 3 (TOE03) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.
  2. To use P30/RTC1HZ/INTP3 as a general-purpose port, set bit 5 (RCLOE1) of Real-time counter control register 0 (RTCC0) to “0”, which is the same as their default status setting.

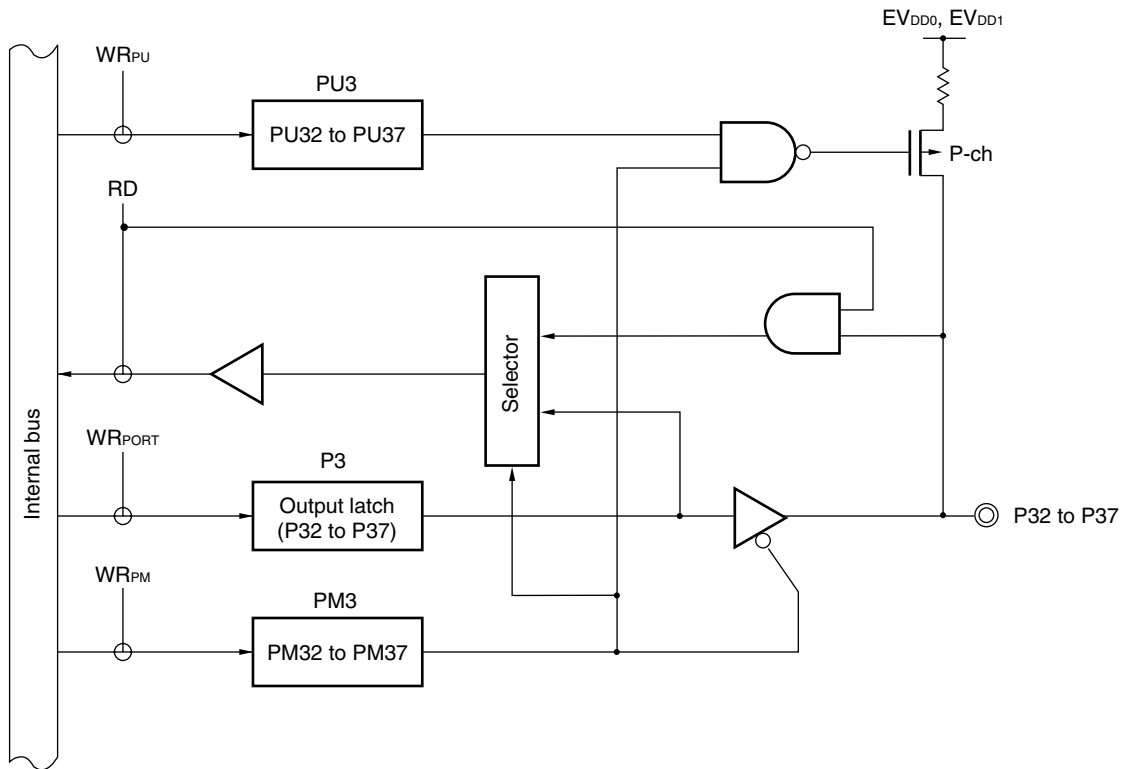
Figure 4-15. Block Diagram of P30 and P31



- P3: Port register 3
- PU3: Pull-up resistor option register 3
- PM3: Port mode register 3
- RD: Read signal
- WR<sub>xx</sub>: Write signal



Figure 4-16. Block Diagram of P32 to P37



- P3: Port register 3
- PU3: Pull-up resistor option register 3
- PM3: Port mode register 3
- RD: Read signal
- $WR_{xx}$ : Write signal

#### 4.2.5 Port 4

Port 4 is an 8-bit I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 to P47 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4)<sup>Note</sup>.

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

Output from the P43 and P45 pins can be specified as N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 4 (POM4).

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

Reset signal generation sets port 4 to input mode.

Figures 4-17 to 4-24 show block diagrams of port 4.

**Note** When a tool is connected, the P40 and P41 pins cannot be connected to a pull-up resistor.

**Cautions 1. When a tool is connected, the P40 pin cannot be used as a port pin.**

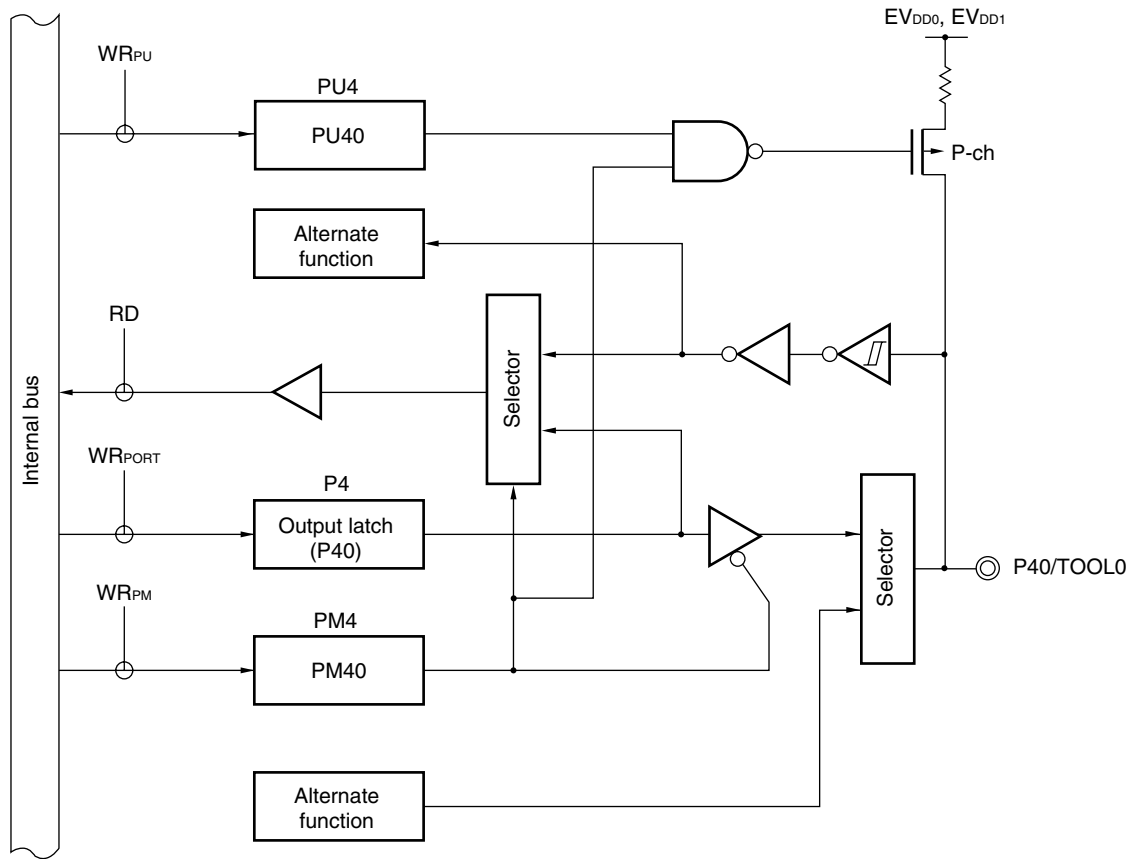
**When the on-chip debug function is used, P41 pin can be used as follows by the mode setting on the debugger.**

**1-line mode: can be used as a port (P41).**

**2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).**

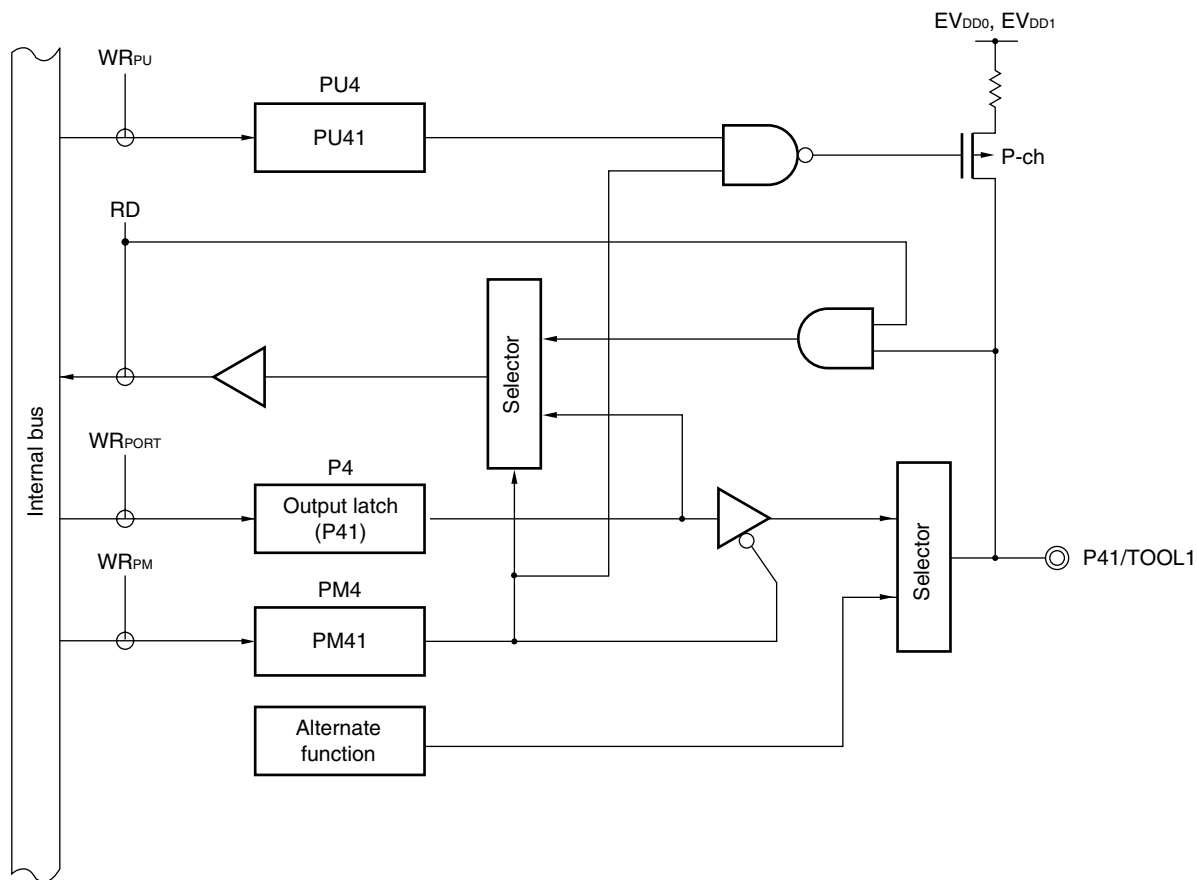
- 2. To use P43/SCK01, P44/SI01 or P45/SO01 as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 13-6 Relationship Between Register Settings and Pins (Channel 1 of Unit 0: CSI01, UART0 reception).**
- 3. To use P42/TI04/TO04 or P46/INTP1/TI05/TO05 as a general-purpose port, set bits 4 and 5 (TO04, TO05) of timer output register 0 (TO0) and bits 4 and 5 (TOE04, TOE05) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.**

Figure 4-17. Block Diagram of P40



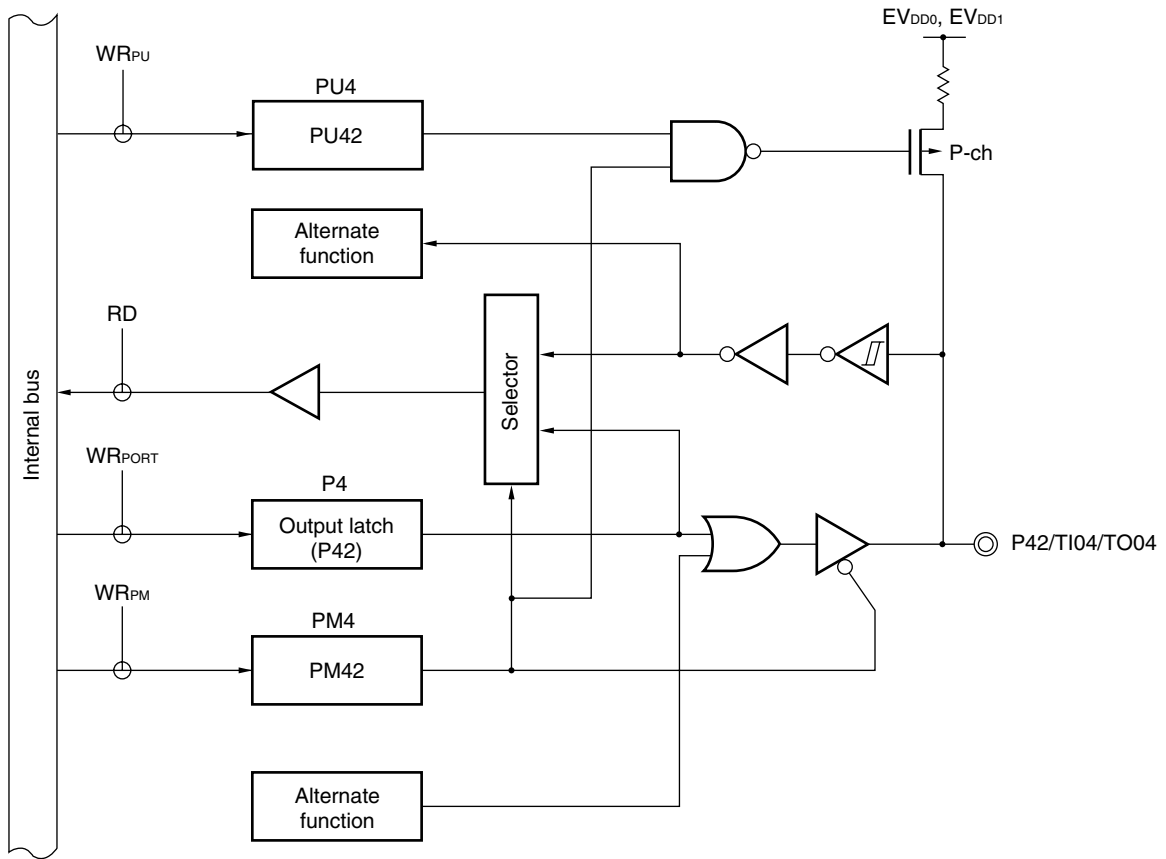
- P4: Port register 4
- PU4: Pull-up resistor option register 4
- PM4: Port mode register 4
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-18. Block Diagram of P41



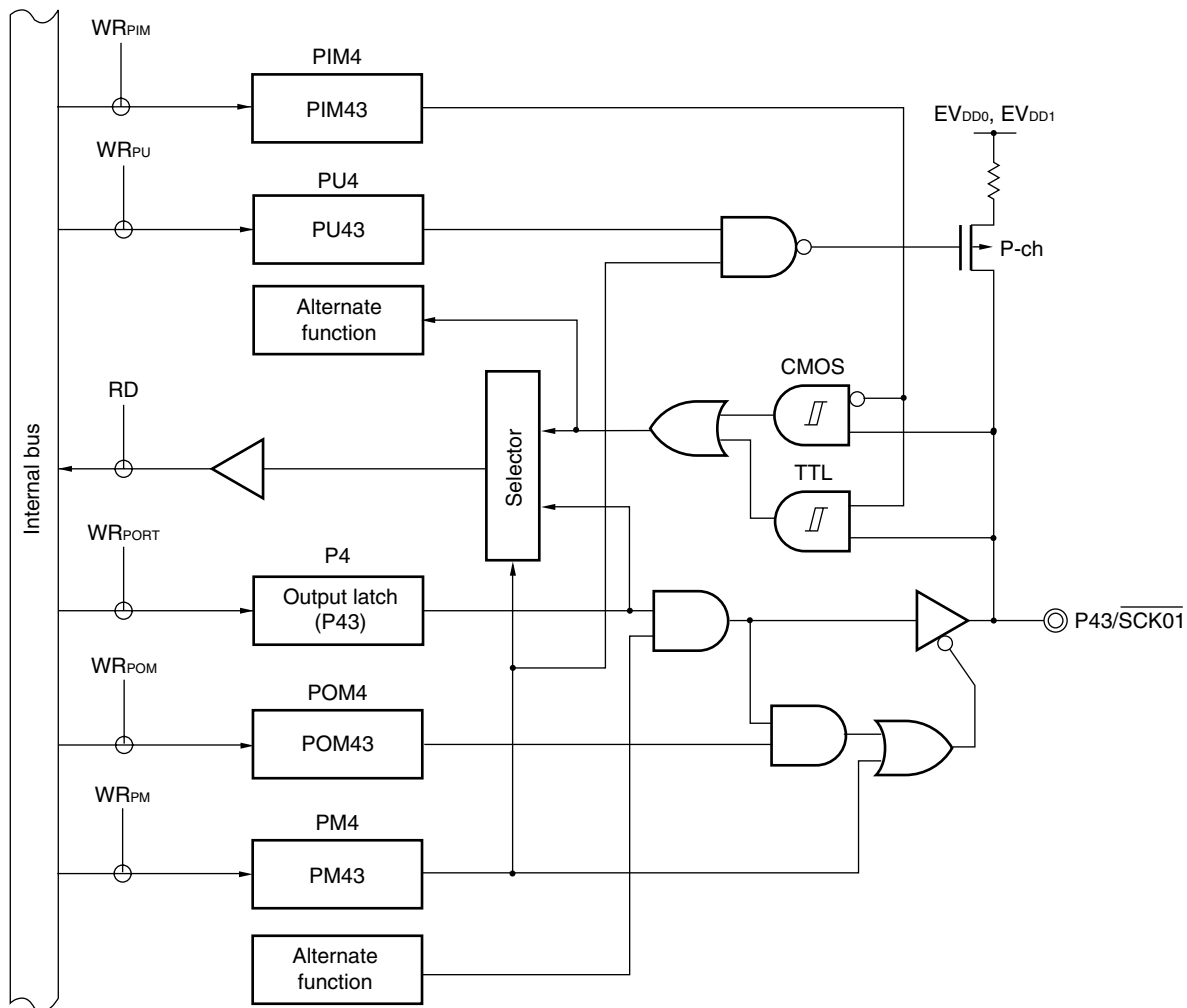
- P4: Port register 4
- PU4: Pull-up resistor option register 4
- PM4: Port mode register 4
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-19. Block Diagram of P42



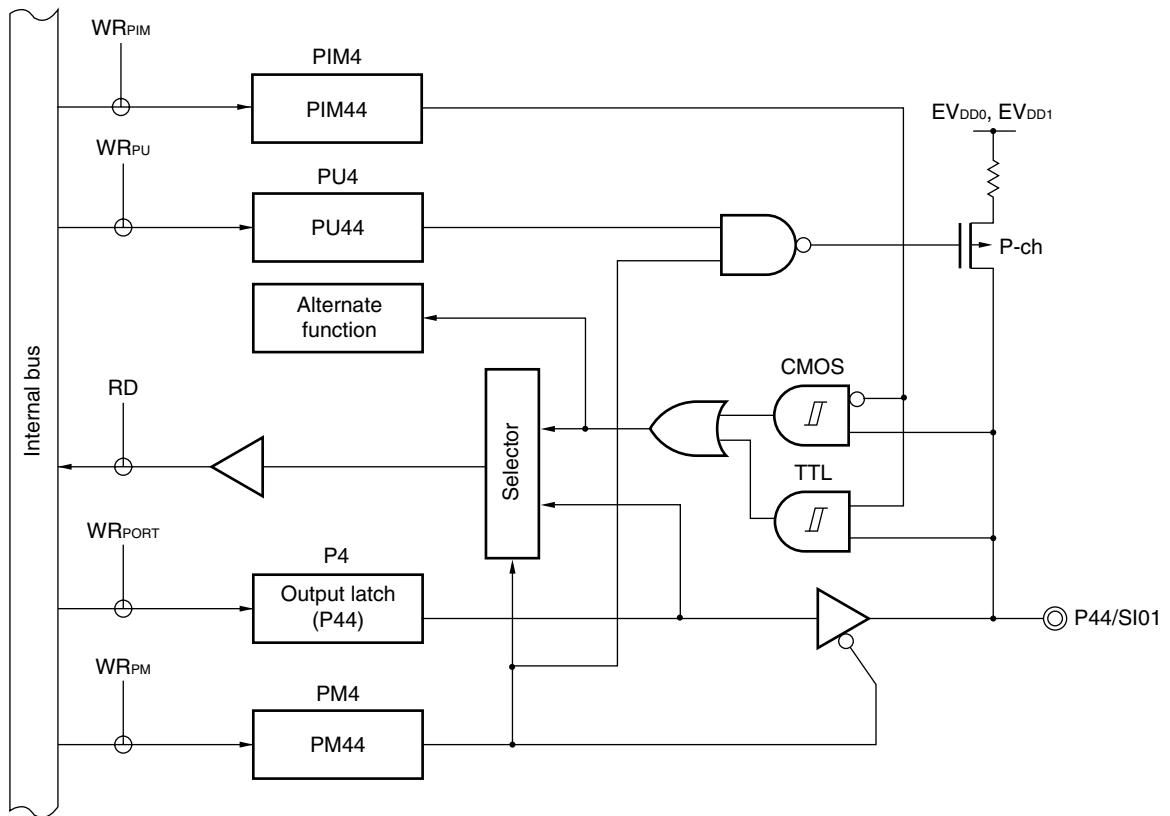
- P4: Port register 4
- PU4: Pull-up resistor option register 4
- PM4: Port mode register 4
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-20. Block Diagram of P43



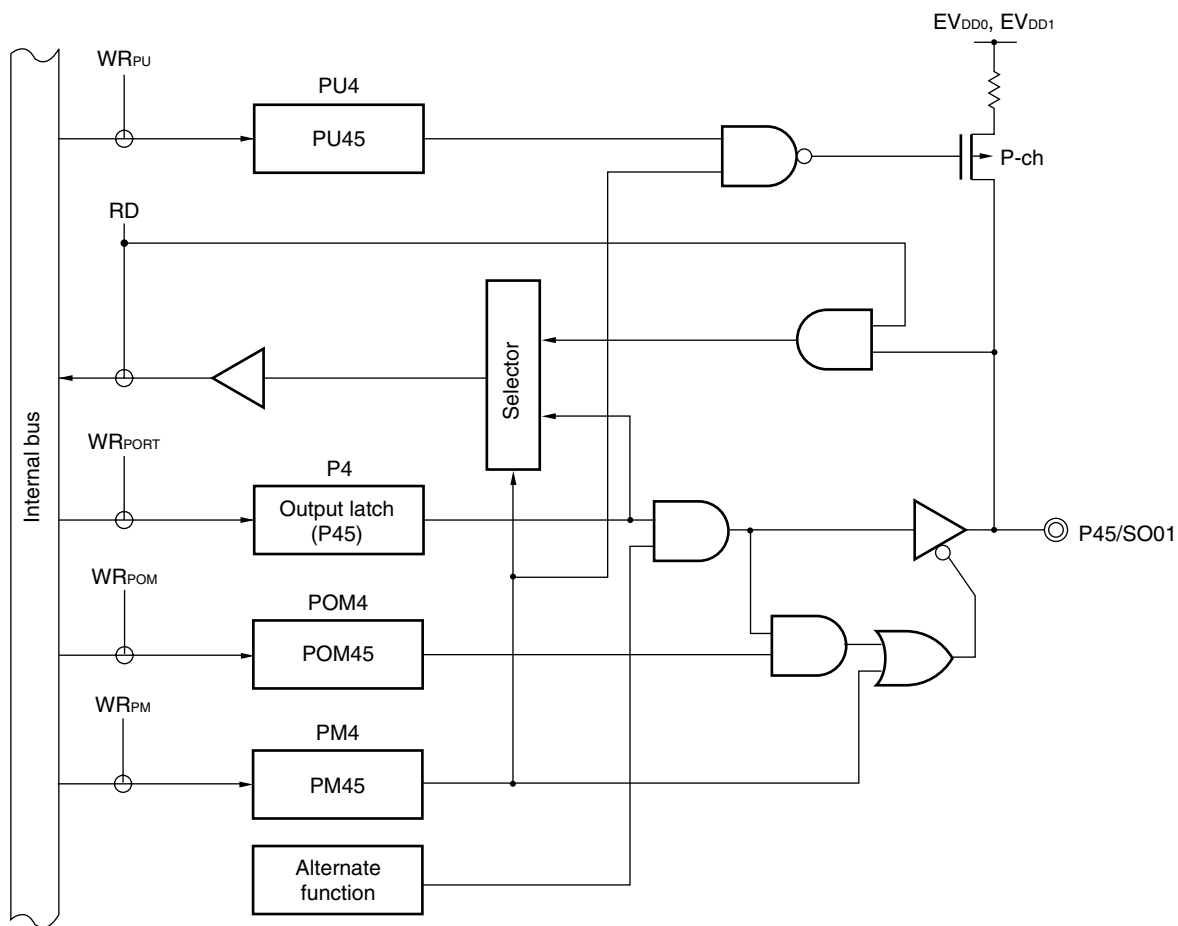
- P4: Port register 4
- PU4: Pull-up resistor option register 4
- PM4: Port mode register 4
- PIM4: Port input mode register 4
- POM4: Port output mode register 4
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-21. Block Diagram of P44



- P4: Port register 4
- PU4: Pull-up resistor option register 4
- PM4: Port mode register 4
- PIM4: Port input mode register 4
- RD: Read signal
- WR<sub>xx</sub>: Write signal

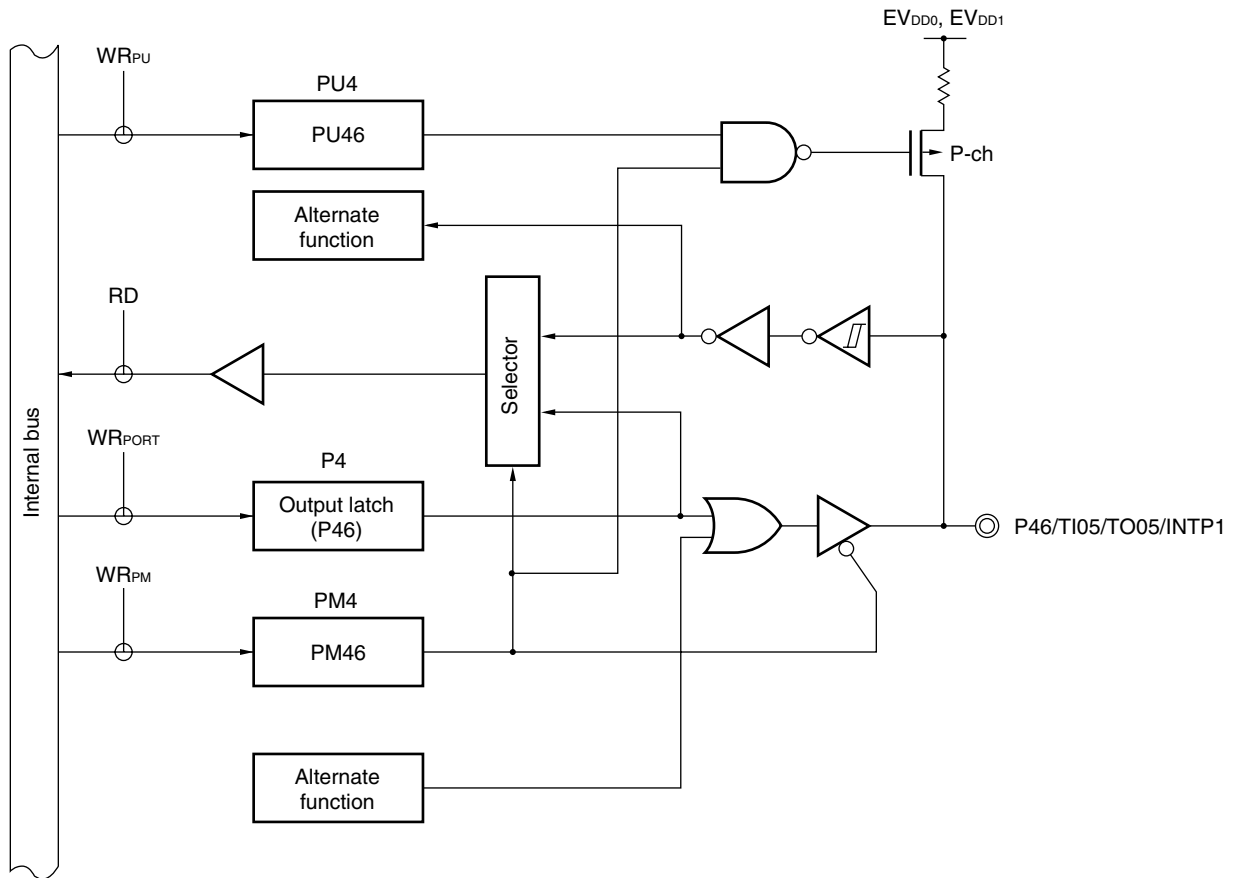
Figure 4-22. Block Diagram of P45



- P4: Port register 4
- PU4: Pull-up resistor option register 4
- PM4: Port mode register 4
- POM4: Port output mode register 4
- RD: Read signal
- WR<sub>xx</sub>: Write signal

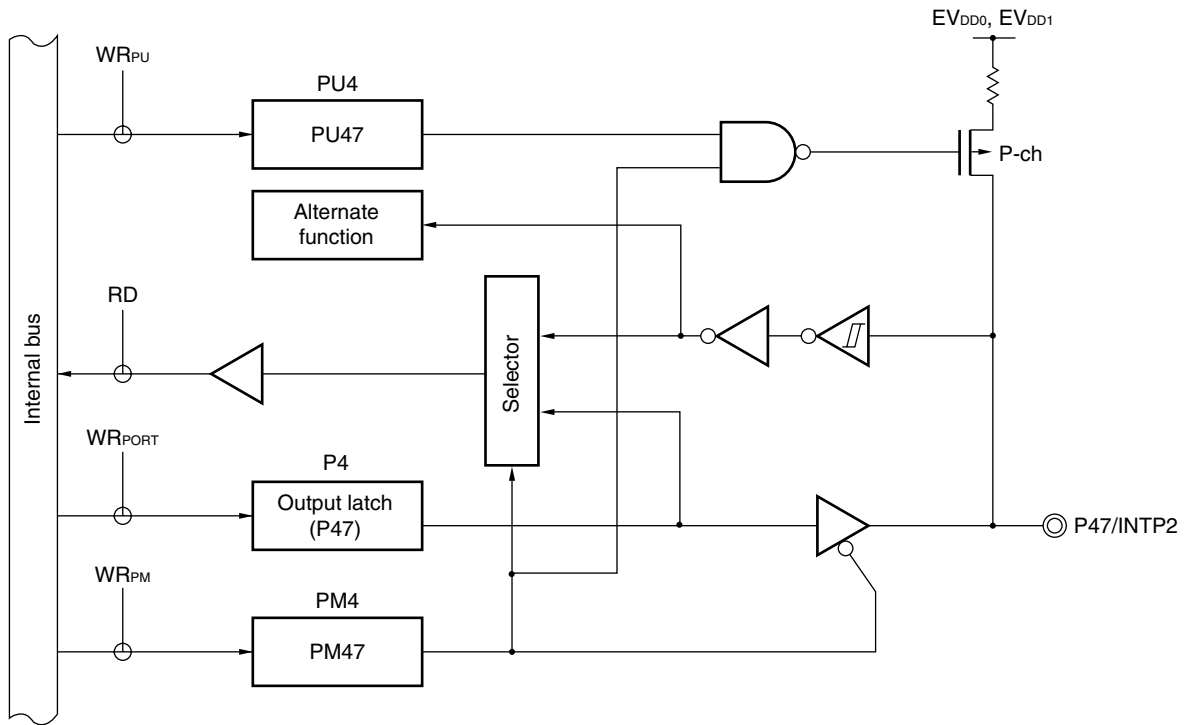


Figure 4-23. Block Diagram of P46



- P4: Port register 4
- PU4: Pull-up resistor option register 4
- PM4: Port mode register 4
- RD: Read signal
- WR<sub>×</sub>: Write signal

Figure 4-24. Block Diagram of P47



- P4: Port register 4
- PU4: Pull-up resistor option register 4
- PM4: Port mode register 4
- RD: Read signal
- WR<sub>xx</sub>: Write signal

4.2.6 Port 5

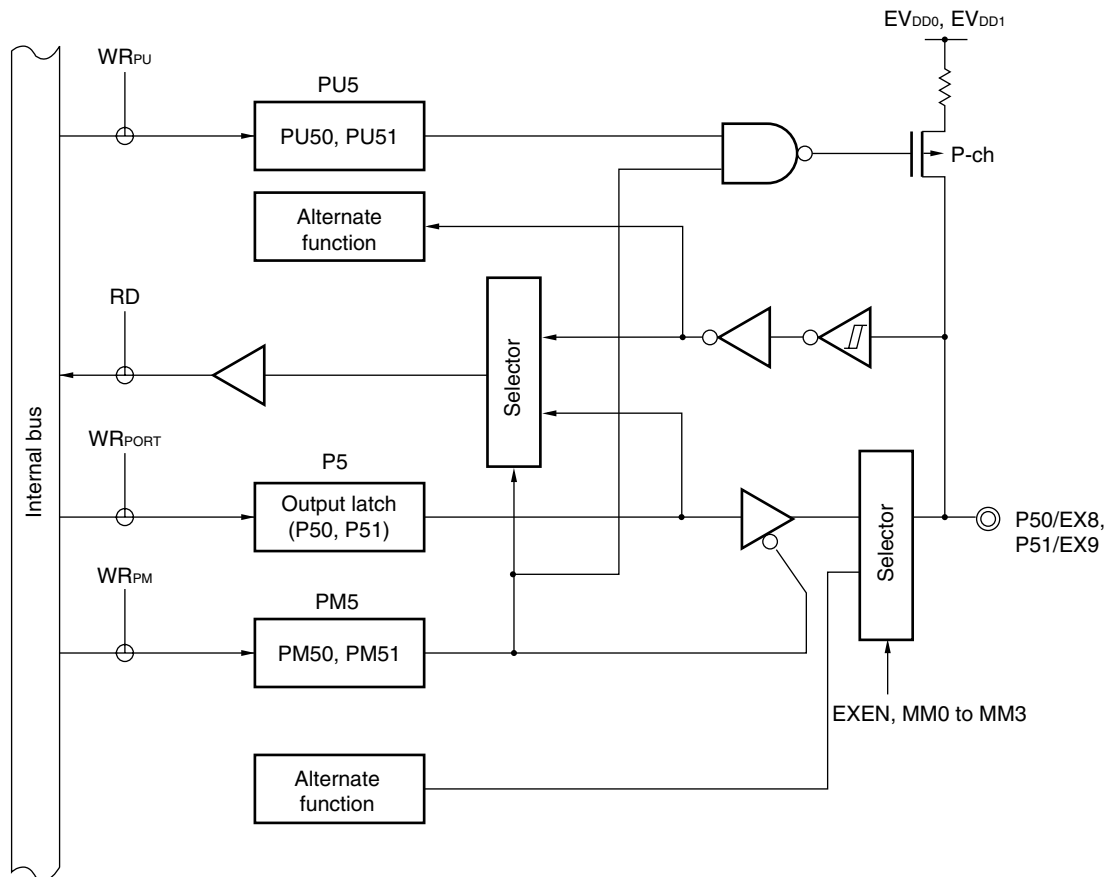
Port 5 is an 8-bit I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P57 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

This port can also be used for external expansion I/O (multiplexed address/data bus, address bus, data bus). When the external expansion I/O function is used, it controls the I/O ignoring the settings on port mode register 5 (PM5), port register 5 (P5), and pull-up resistor option register 5 (PU5).

Reset signal generation sets port 5 to input mode.

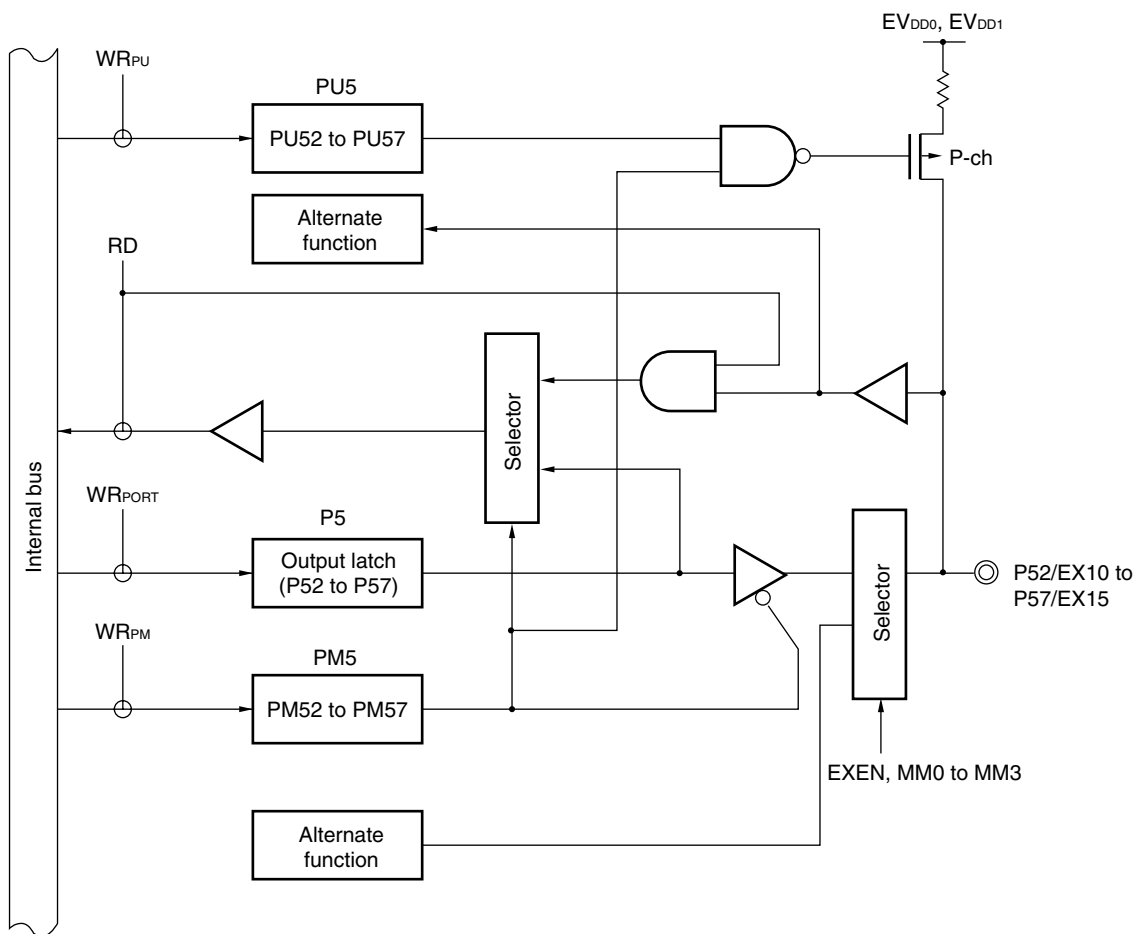
Figures 4-25 and 4-26 show block diagrams of port 5.

Figure 4-25. Block Diagram of P50 and P51



- P5: Port register 5
- PU5: Pull-up resistor option register 5
- PM5: Port mode register 5
- RD: Read signal
- WR<sub>xx</sub>: Write signal
- EXEN: Bit 7 of memory extension mode control register (MEM)
- MM0 to MM3: Bits 0 to 3 of memory extension mode control register (MEM)

Figure 4-26. Block Diagram of P52 to P57



- P5: Port register 5
- PU5: Pull-up resistor option register 5
- PM5: Port mode register 5
- RD: Read signal
- WR<sub>xx</sub>: Write signal
- EXEN: Bit 7 of memory extension mode control register (MEM)
- MM0 to MM3: Bits 0 to 3 of memory extension mode control register (MEM)

4.2.7 Port 6

Port 6 is an 8-bit I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6). When the P64 to P67 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 6 (PU6).

The output of the P60 to P63 pins is N-ch open-drain output (6 V tolerance).

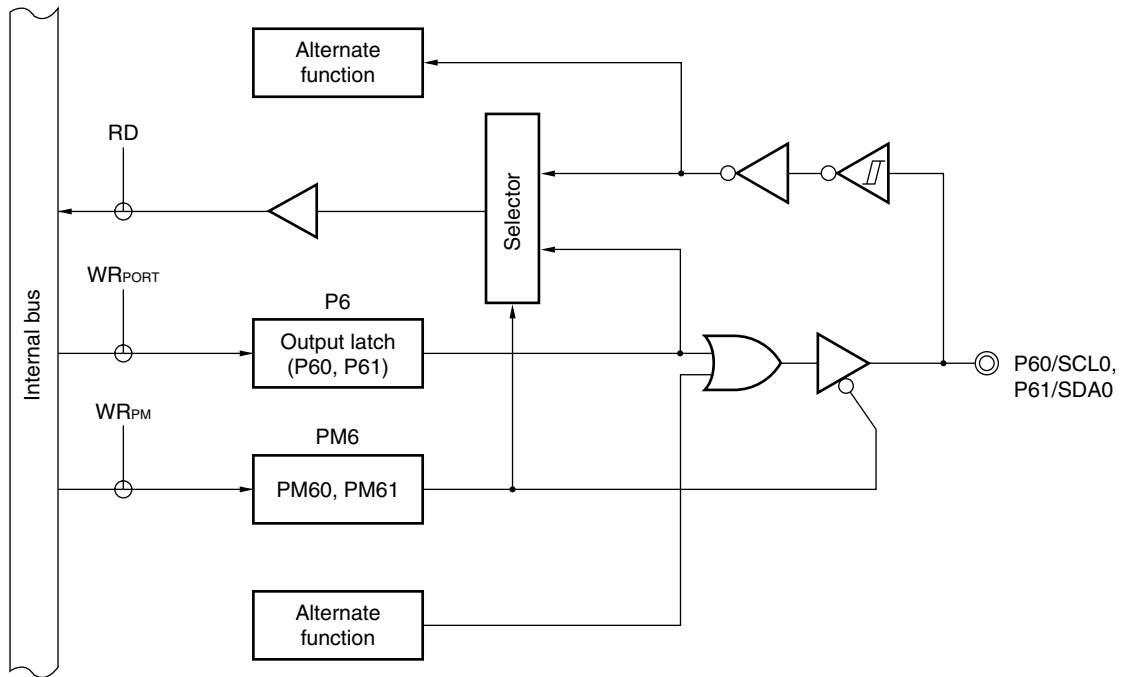
This port can also be used for serial interface data I/O, clock I/O, read strobe signal output, write strobe signal output (8-bit bus mode, 16-bit bus mode (lower)), write strobe signal output (16-bit bus mode (higher)), and address strobe signal output.

Reset signal generation sets port 6 to input mode.

Figures 4-27 to 4-29 show block diagrams of port 6.

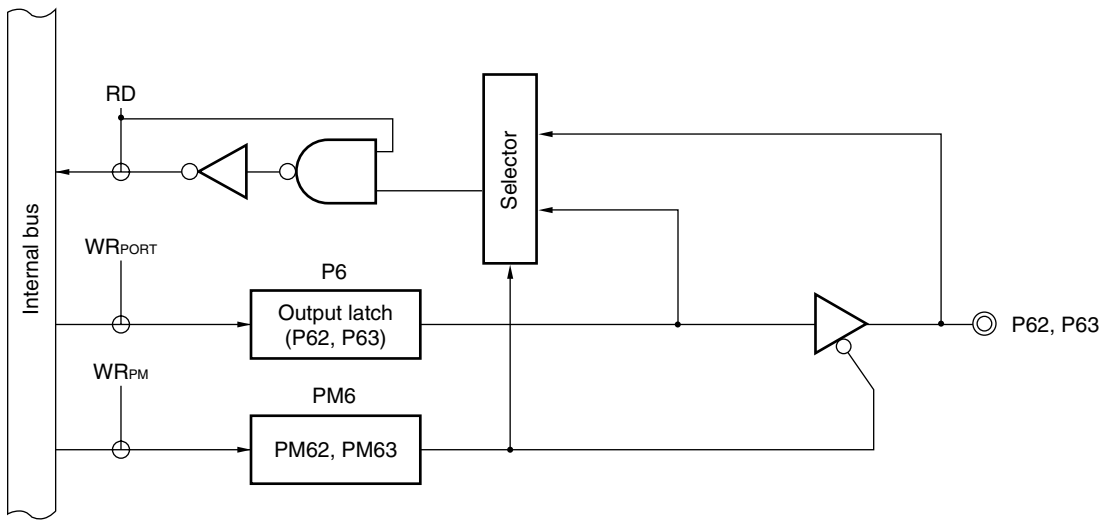
**Caution** Stop the operation of serial interface IIC0 when using P60/SCL0 and P61/SDA0 as general-purpose ports.

Figure 4-27. Block Diagram of P60 and P61



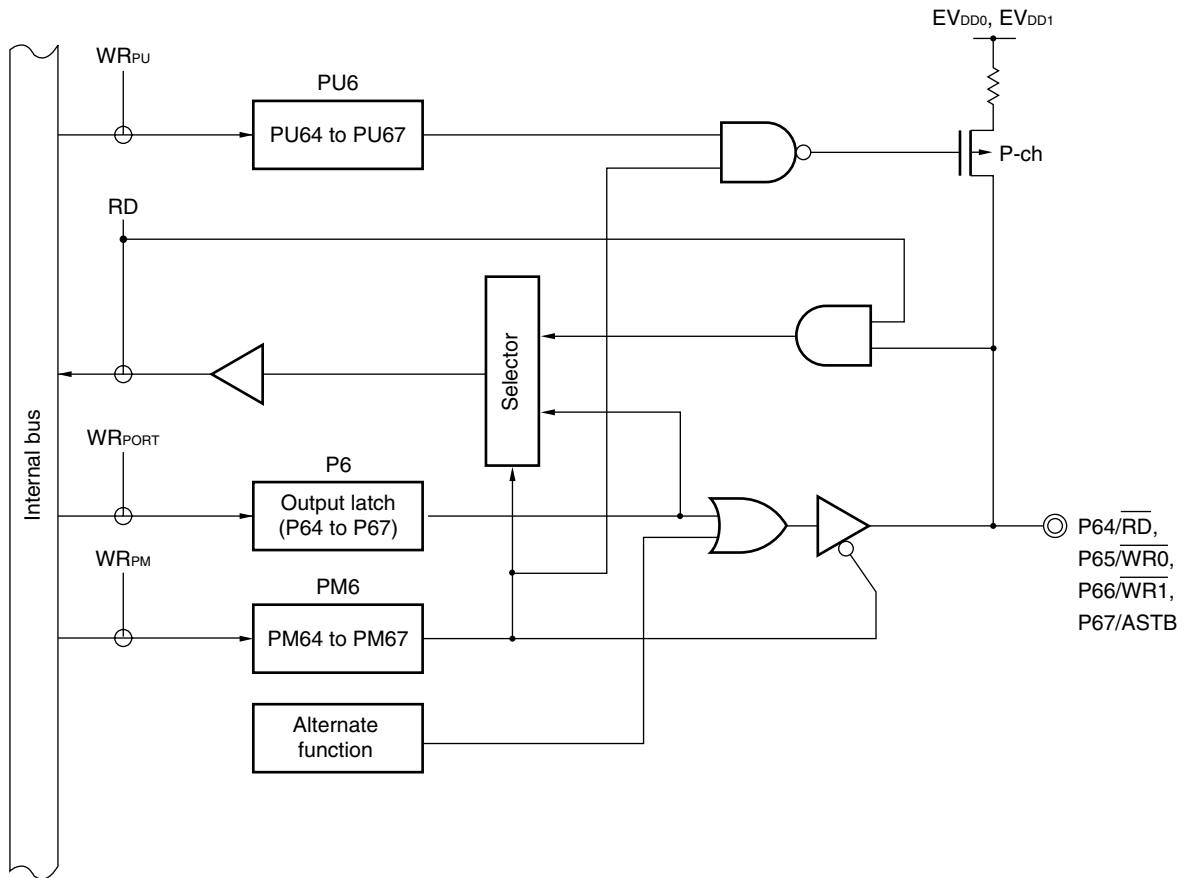
- P6: Port register 6
- PM6: Port mode register 6
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-28. Block Diagram of P62 and P63



- P6: Port register 6
- PM6: Port mode register 6
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-29. Block Diagram of P64 to P67



- P6: Port register 6
- PU6: Pull-up resistor option register 6
- PM6: Port mode register 6
- RD: Read signal
- $WR_{xx}$ : Write signal

4.2.8 Port 7

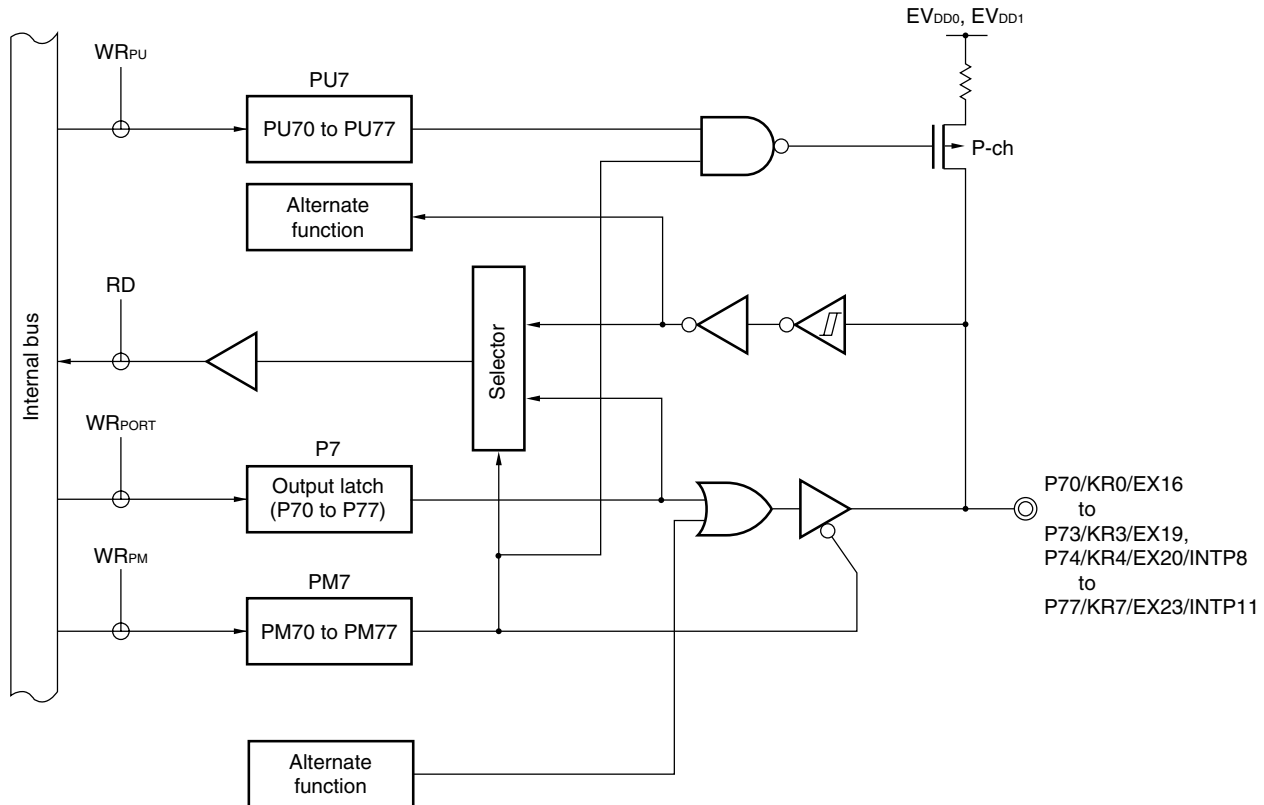
Port 7 is an 8-bit I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

This port can also be used for key return input, external expansion output (address bus), and interrupt request input.

Reset signal generation sets port 7 to input mode.

Figure 4-30 shows a block diagram of port 7.

Figure 4-30. Block Diagram of P70 to P77



- P7: Port register 7
- PU7: Pull-up resistor option register 7
- PM7: Port mode register 7
- RD: Read signal
- WR<sub>xx</sub>: Write signal



4.2.9 Port 8

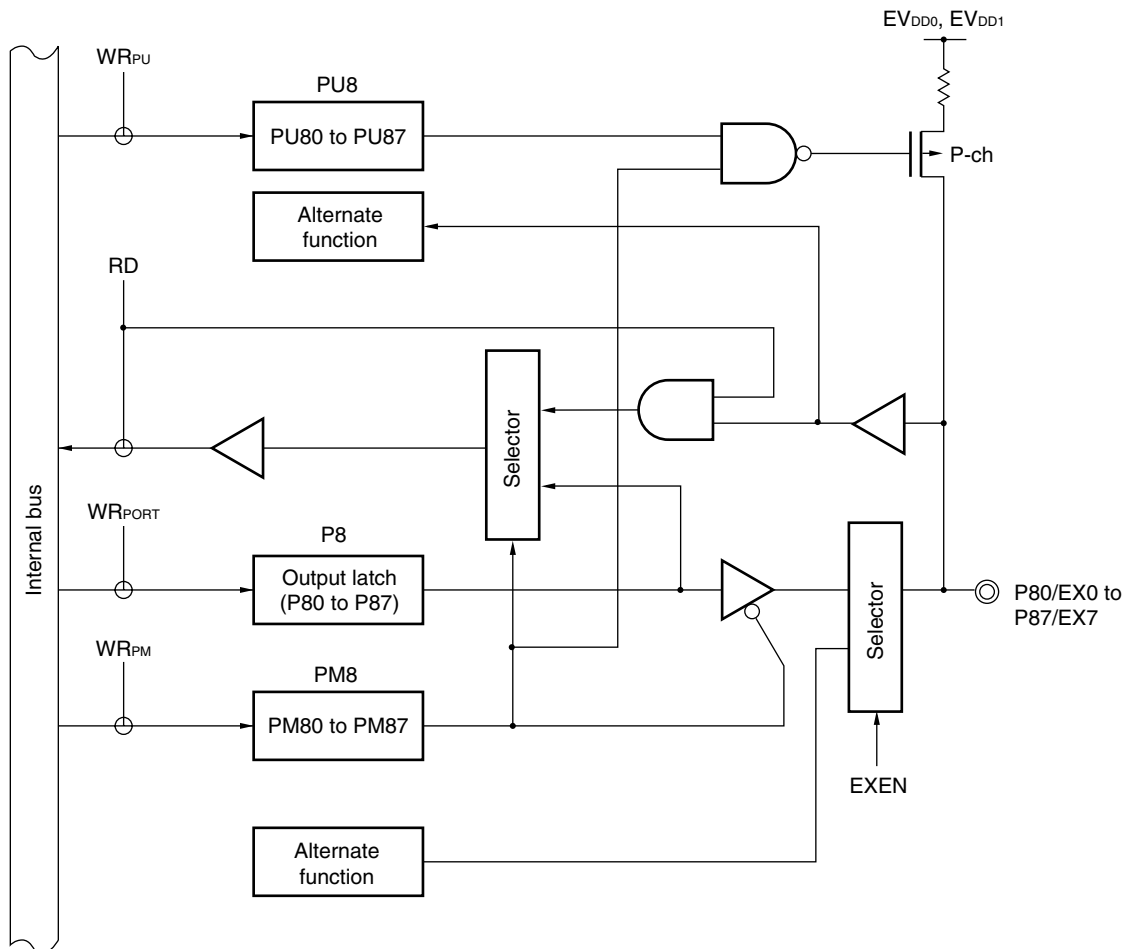
Port 8 is an 8-bit I/O port with an output latch. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8). When the P80 to P87 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 8 (PU8).

This port can also be used for external expansion I/O (multiplexed address/data bus, data bus). When the external expansion I/O function is used, it controls the I/O ignoring the settings on port mode register 8 (PM8), port register 8 (P8), and pull-up resistor option register 8 (PU8).

Reset signal generation sets port 8 to input mode.

Figure 4-31 shows a block diagram of port 8.

Figure 4-31. Block Diagram of P80 to P87



- P8: Port register 8
- PU8: Pull-up resistor option register 8
- PM8: Port mode register 8
- RD: Read signal
- WR<sub>xx</sub>: Write signal
- EXEN: Bit 7 of memory extension mode control register (MEM)

4.2.10 Port 9

Port 9 is an 8-bit I/O port with an output latch. Port 9 can be set to the input mode or output mode in 1-bit units using port mode register 9 (PM9). When the P90 to P97 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 9 (PU9).

Input to the P95 and P96 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 9 (PIM9).

Output from the P95 to P97 pins can be specified as N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 9 (POM9).

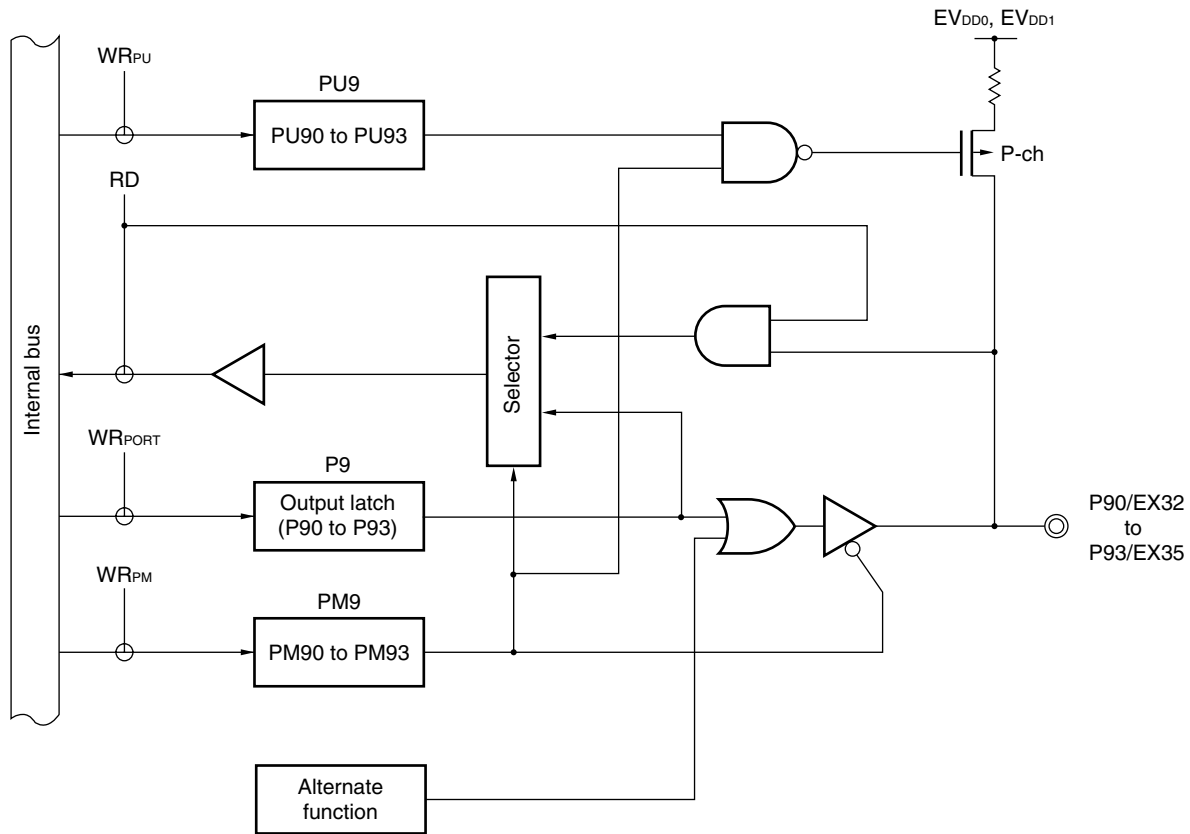
This port can also be used for external expansion output (address bus), serial interface data I/O, and clock I/O

Reset signal generation sets port 0 to input mode.

Figures 4-32 to 4-35 show block diagrams of port 0.

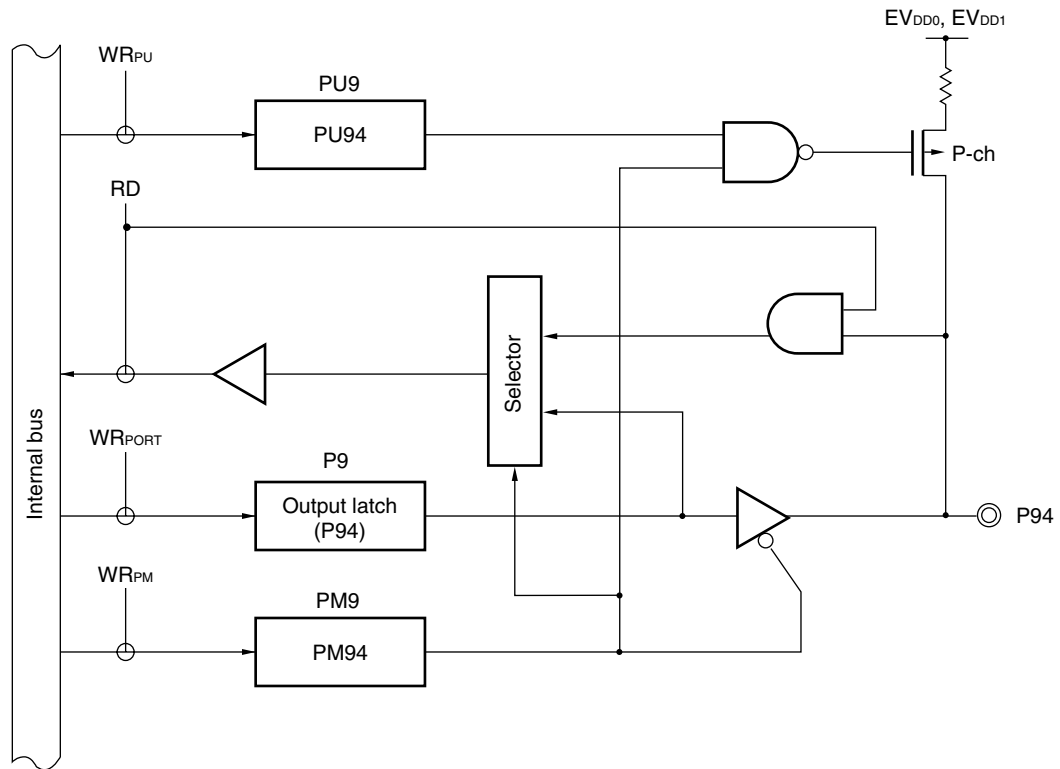
**Caution** To use P95/SCK11/SCL11, P96/SI11/SDA11, or P97/SO11 as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 13-8 Relationship Between Register Settings and Pins (Channel 3 of Unit 0: CSI11, UART1 Reception, IIC11).

Figure 4-32. Block Diagram of P90 to P93



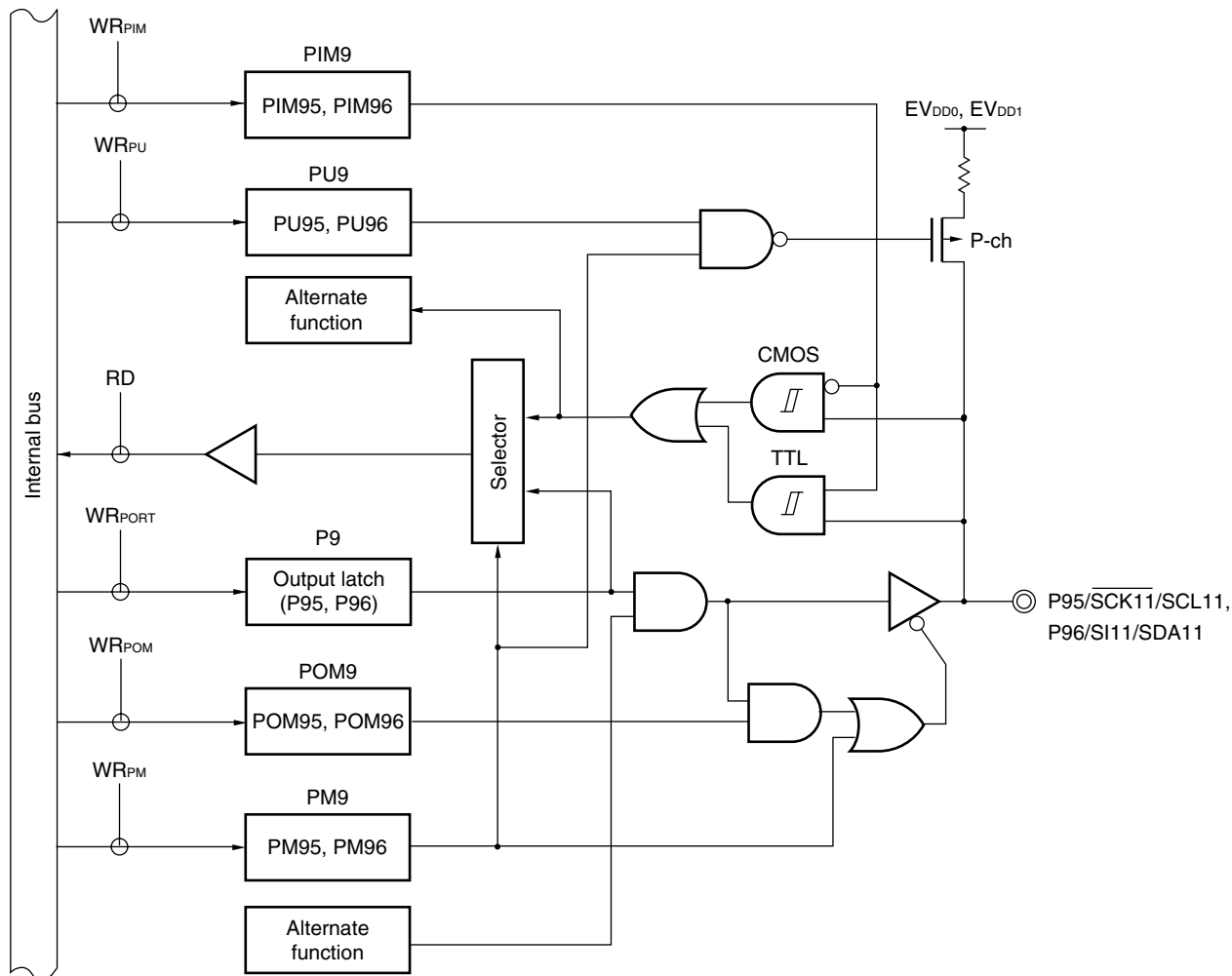
- P9: Port register 9
- PU9: Pull-up resistor option register 9
- PM9: Port mode register 9
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-33. Block Diagram of P94



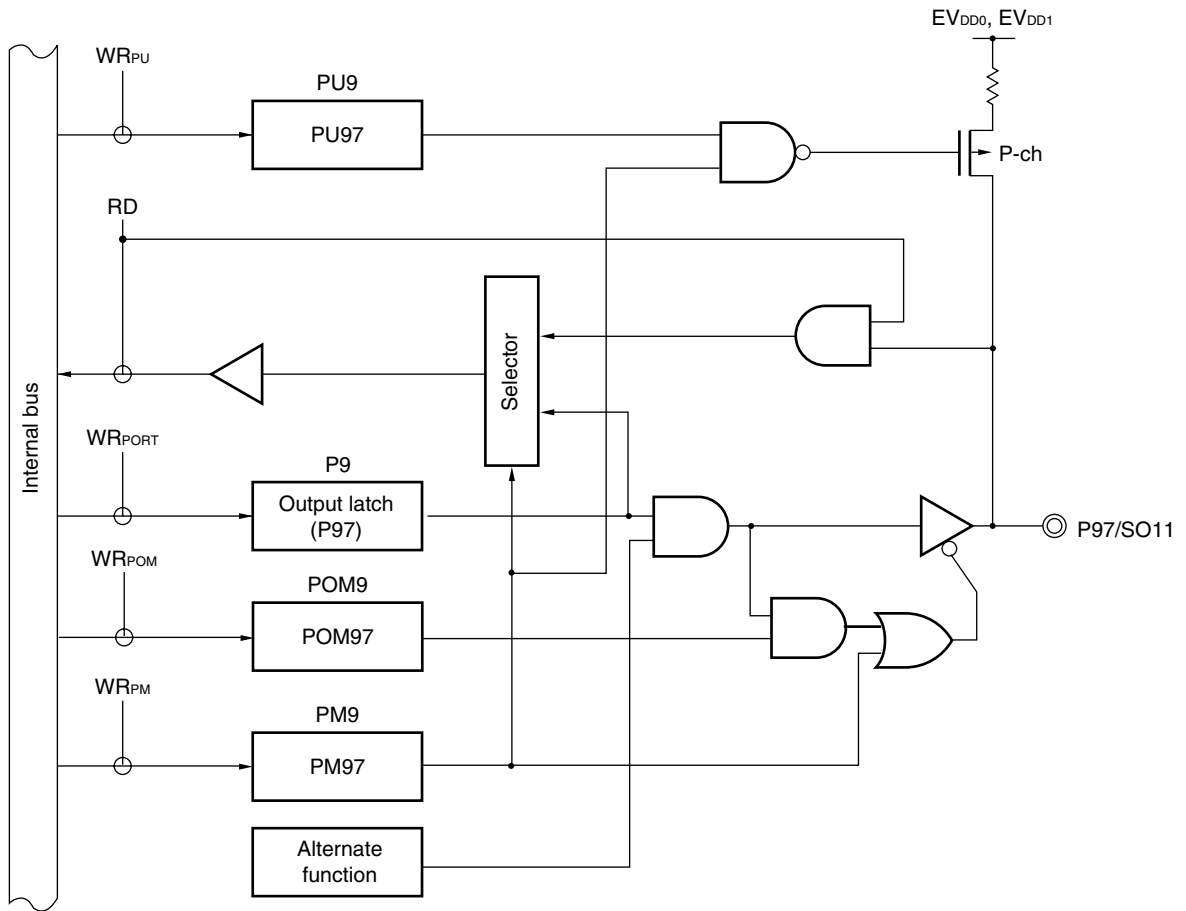
- P9: Port register 9
- PU9: Pull-up resistor option register 9
- PM9: Port mode register 9
- RD: Read signal
- $WR_{xx}$ : Write signal

Figure 4-34. Block Diagram of P95 and P96



- P9: Port register 9
- PU9: Pull-up resistor option register 9
- PM9: Port mode register 9
- PIM9: Port input mode register 9
- POM9: Port output mode register 9
- RD: Read signal
- WR<sub>xx</sub>: Write signal

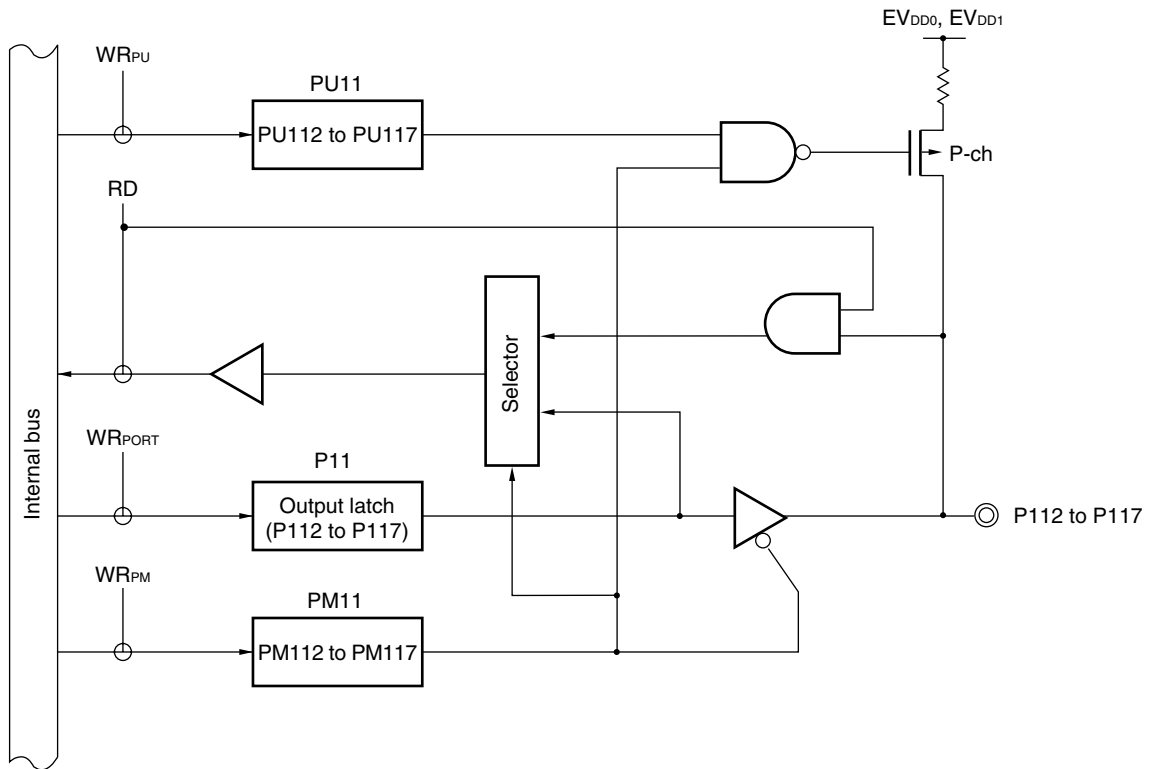
Figure 4-35. Block Diagram of P97



- P9: Port register 9
- PU9: Pull-up resistor option register 9
- PM9: Port mode register 9
- POM9: Port output mode register 9
- RD: Read signal
- $WR_{xx}$ : Write signal



Figure 4-37. Block Diagram of P112 to P117



- P11: Port register 11
- PU11: Pull-up resistor option register 11
- PM11: Port mode register 11
- RD: Read signal
- $WR_{xx}$ : Write signal

4.2.12 Port 12

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

P121 to P124 are 4-bit input ports.

Input to the P125 and P126 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 12 (PIM12).

Output from the P125 to P127 pins can be specified as N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 12 (POM12).

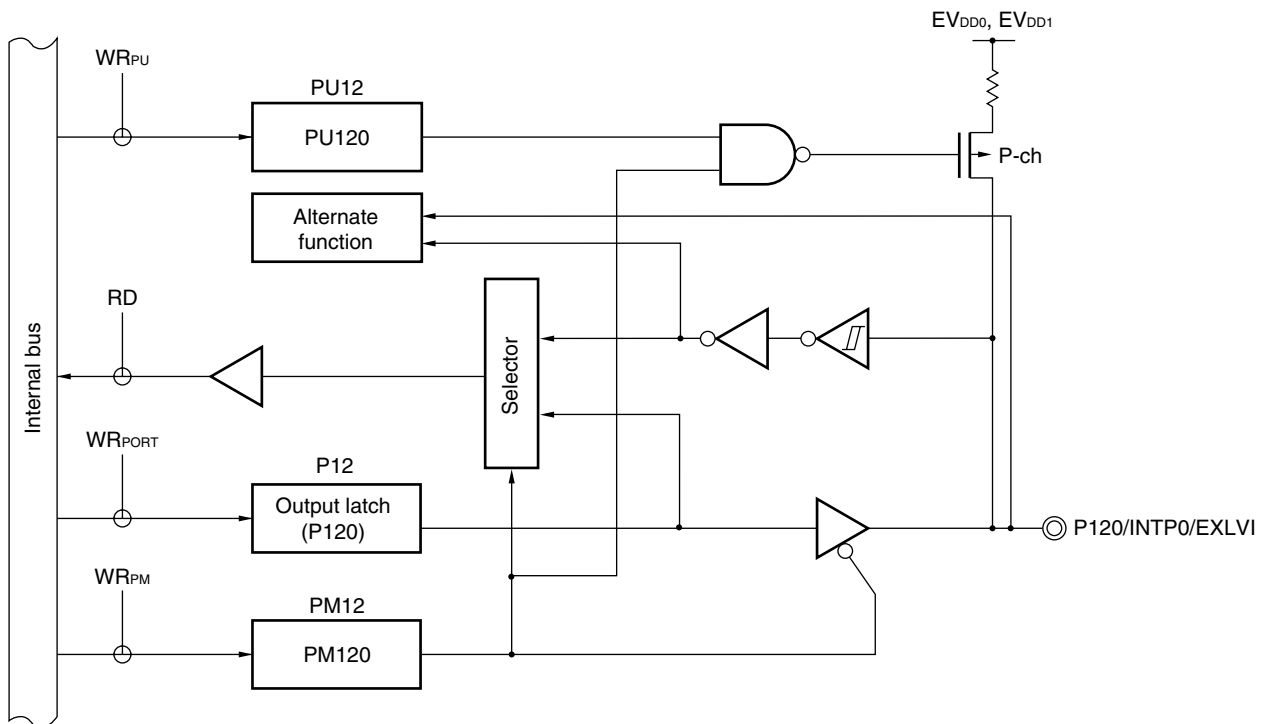
This port can also be used for external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, serial interface data I/O, and clock I/O.

Reset signal generation sets port 12 to input mode.

Figures 4-38 to 4-42 show block diagrams of port 12.

- Caution 1.** The function setting on P121 to P124 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.
- 2.** To use P125/SCK21/SCL21, P126/SI21/SDA21, or P127/SO21 as a general-purpose port, note the serial array unit 1 setting. For details, refer to Table 13-10 Relationship Between Register Settings and Pins (Channel 1 of Unit 1: CSI21, UART2 reception, IIC21).

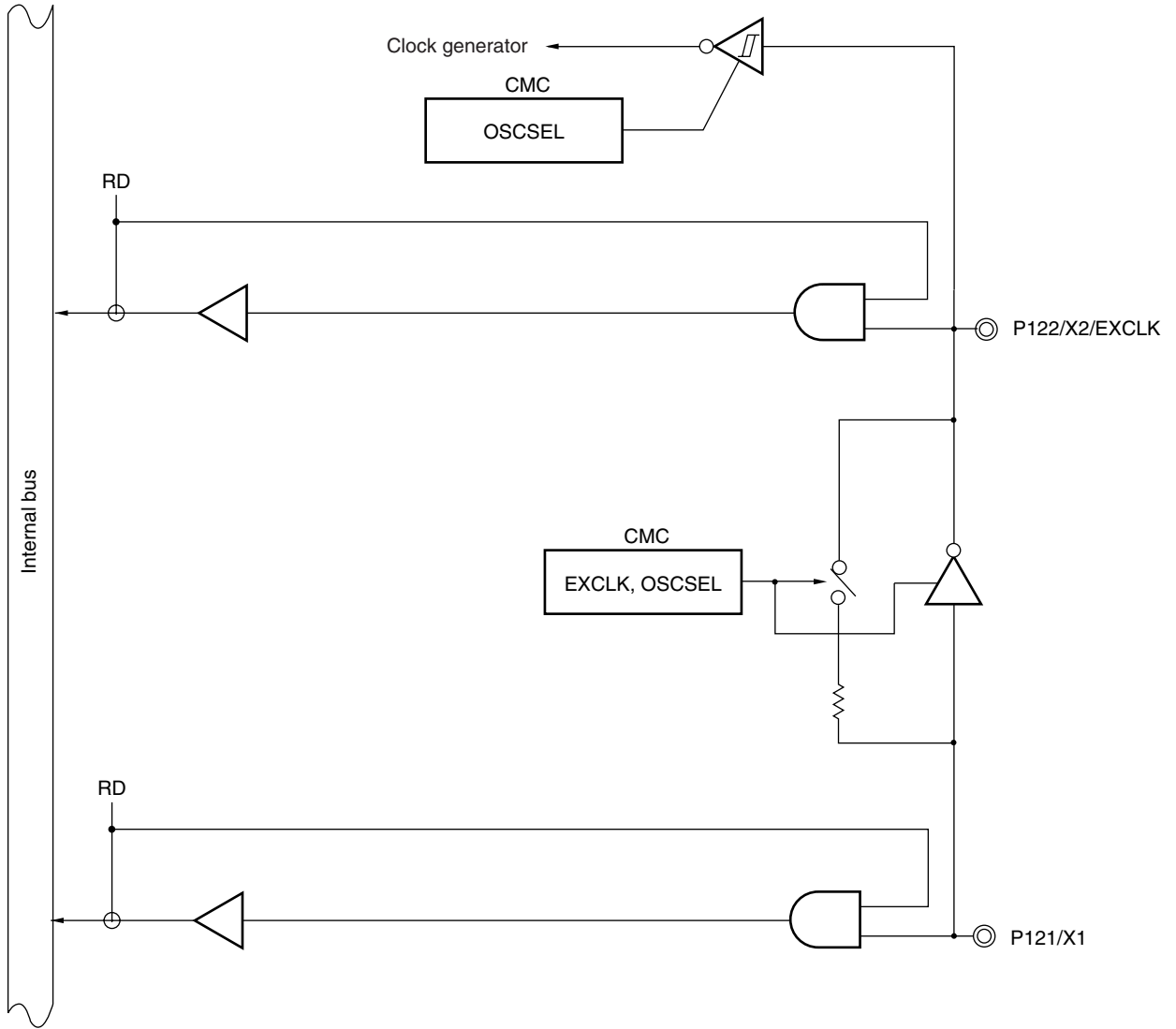
Figure 4-38. Block Diagram of P120



- P12: Port register 12
- PU12: Pull-up resistor option register 12
- PM12: Port mode register 12
- RD: Read signal
- WR<sub>xx</sub>: Write signal

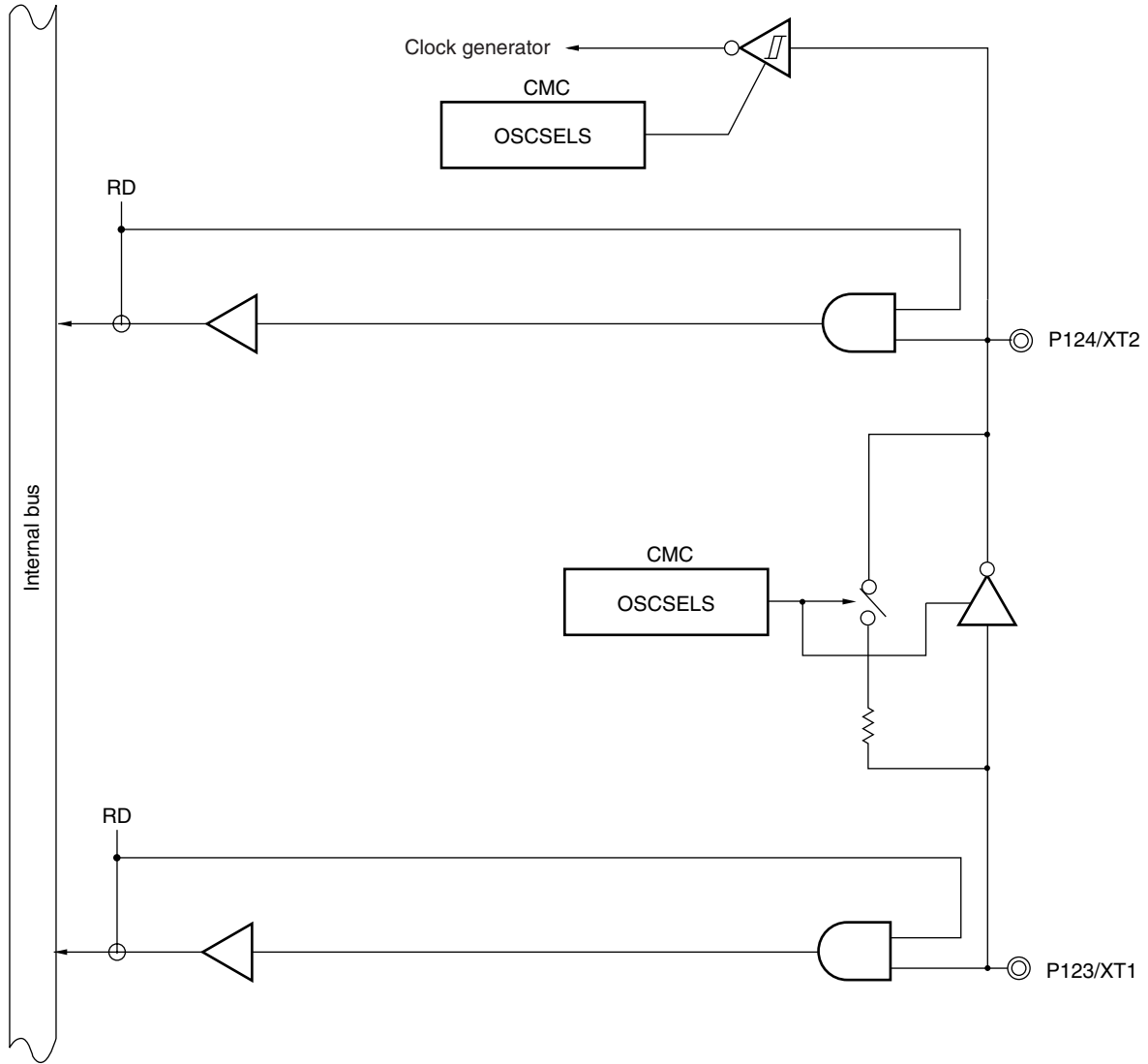


Figure 4-39. Block Diagram of P121 and P122



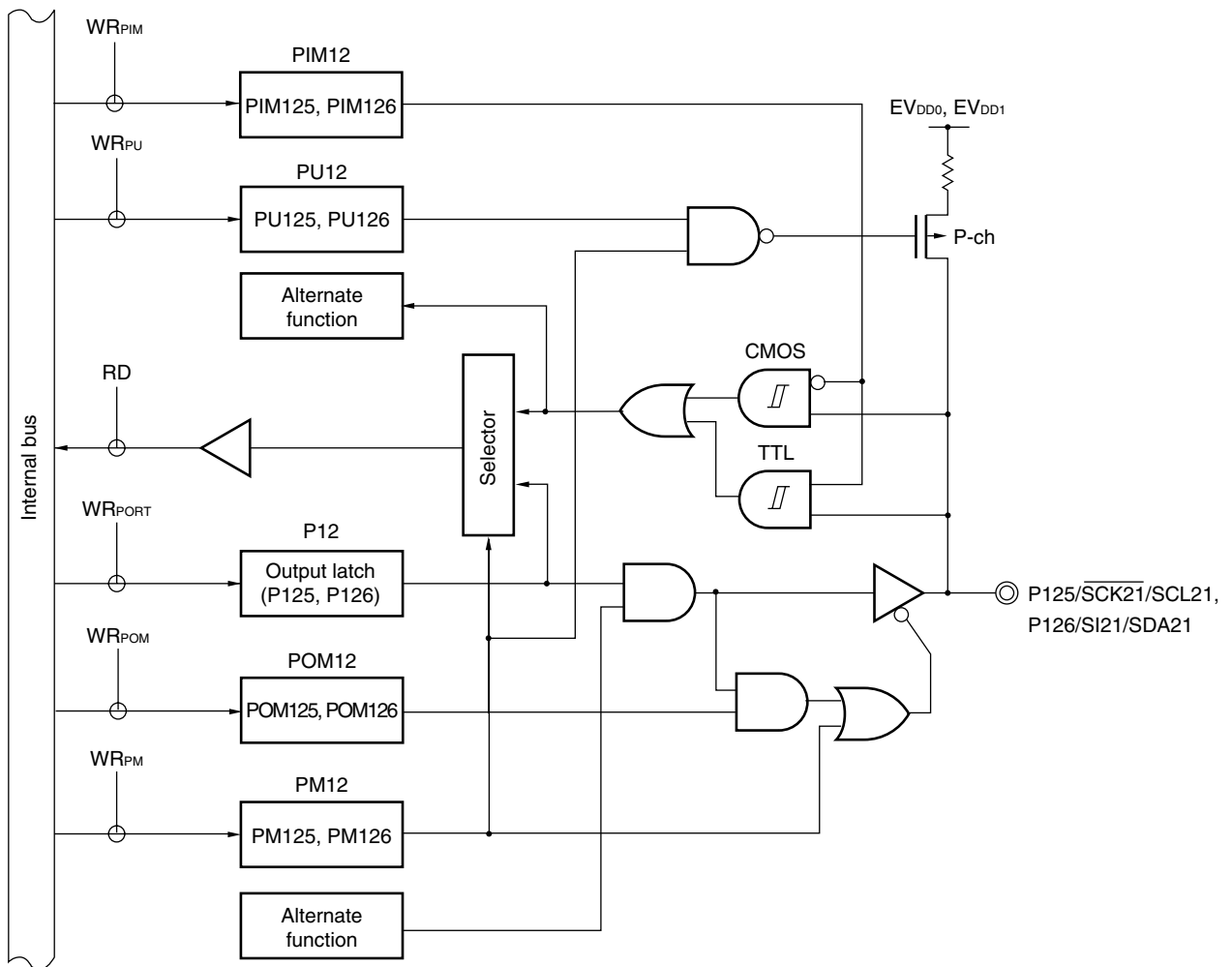
CMC: Clock operation mode control register  
 RD: Read signal

Figure 4-40. Block Diagram of P123 and P124



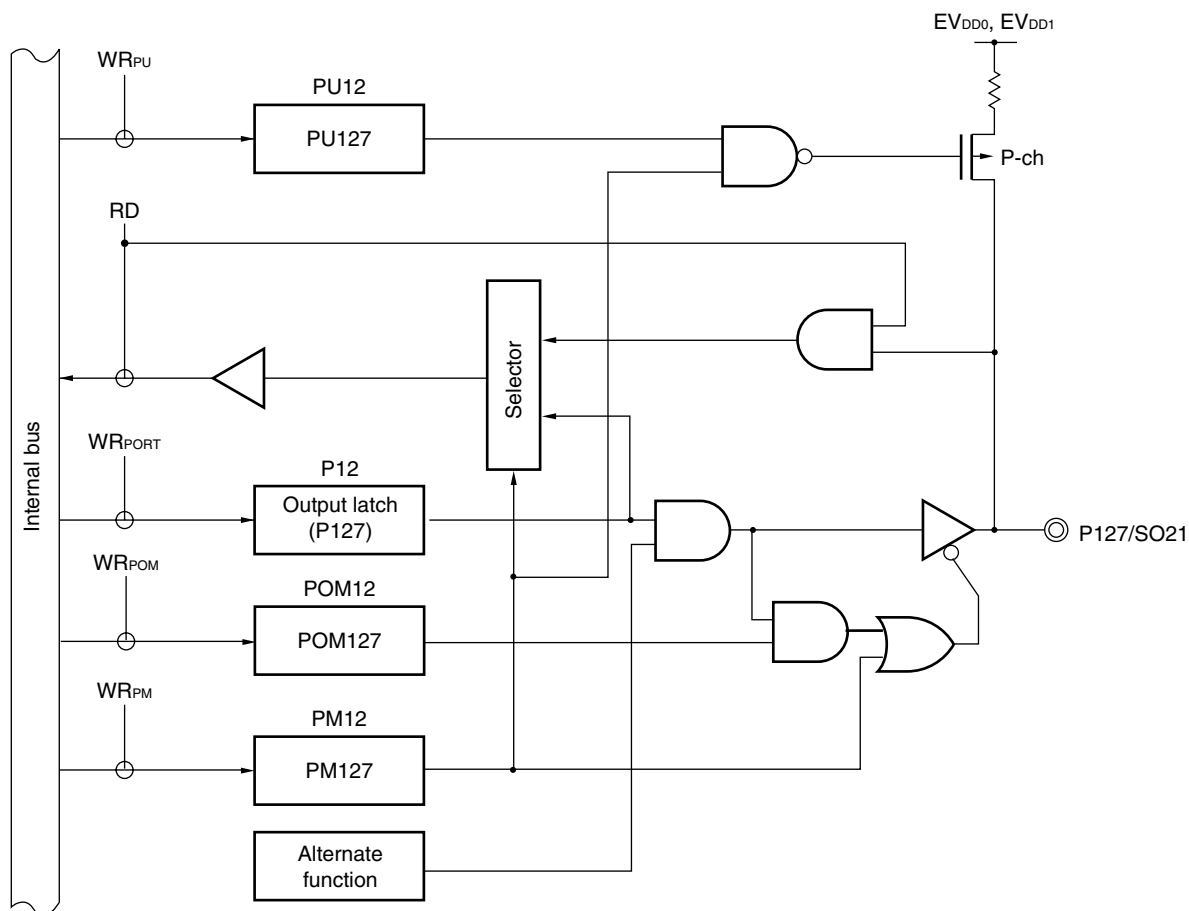
CMC: Clock operation mode control register  
 RD: Read signal

Figure 4-41. Block Diagram of P125 and P126



- P12: Port register 12
- PU12: Pull-up resistor option register 12
- PM12: Port mode register 12
- PIM12: Port input mode register 12
- POM12: Port output mode register 12
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-42. Block Diagram of P127



- P12: Port register 12
- PU12: Pull-up resistor option register 12
- PM12: Port mode register 12
- POM12: Port output mode register 12
- RD: Read signal
- WR<sub>xx</sub>: Write signal

4.2.13 Port 13

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

P131 is a 1-bit I/O port with an output latch. When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 13 (PU13).

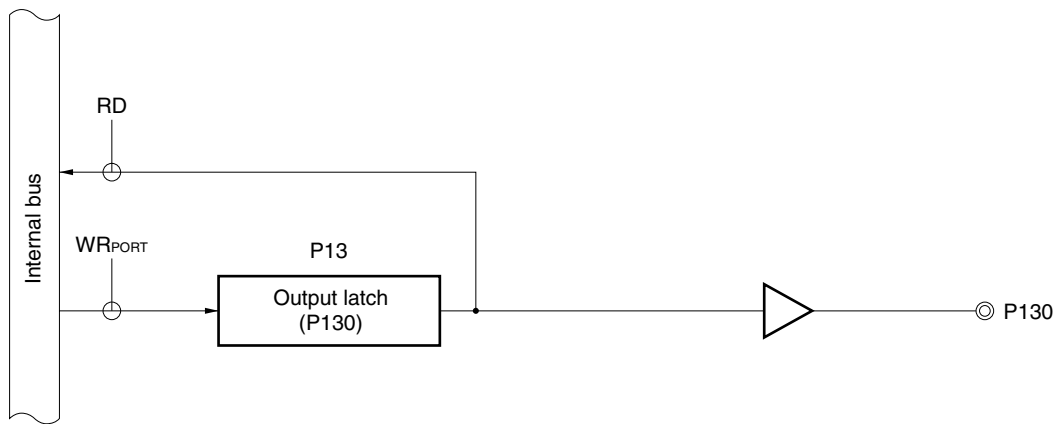
Reset signal generation sets port 13 to input mode.

This port can also be used for timer I/O.

Figures 4-43 and 4-44 show block diagrams of port 13.

**Caution** To use P131/TI06/TO06 as a general-purpose port, set bit 6 (TO06) of timer output register 0 (TO0) and bit 6 (TOE06) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.

Figure 4-43. Block Diagram of P130



P13: Port register 13  
 RD: Read signal  
 WR<sub>xx</sub>: Write signal

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

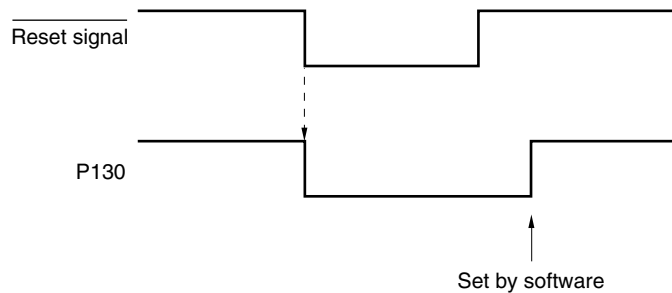
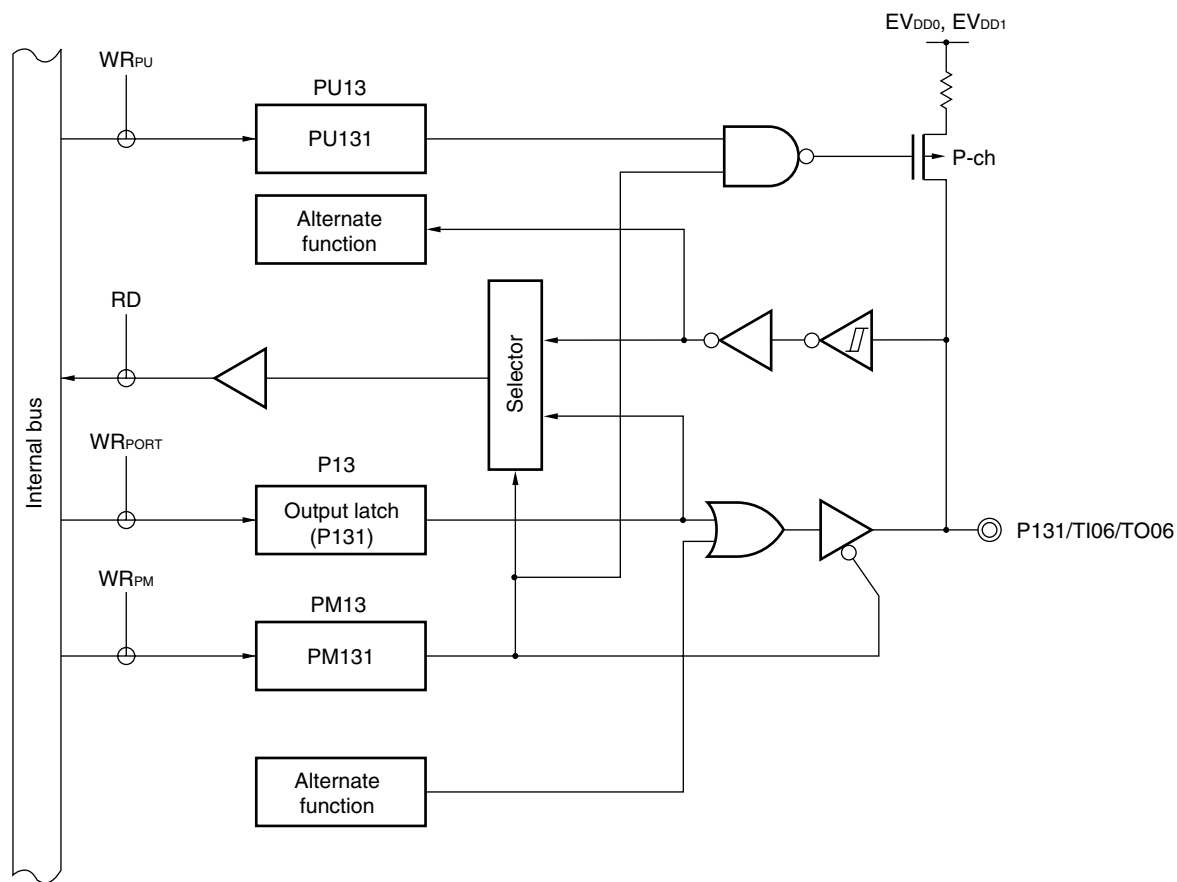


Figure 4-44. Block Diagram of P131



- P13: Port register 13
- PU13: Pull-up resistor option register 13
- PM13: Port mode register 13
- RD: Read signal
- WR<sub>xx</sub>: Write signal

#### 4.2.14 Port 14

Port 14 is a 6-bit I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140 to P145 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 14 (PU14).

Input to the P142 and P143 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 14 (PIM14).

Output from the P142 to P144 pins can be specified as N-ch open-drain output ( $V_{DD}$  tolerance) in 1-bit units using port output mode register 14 (POM14).

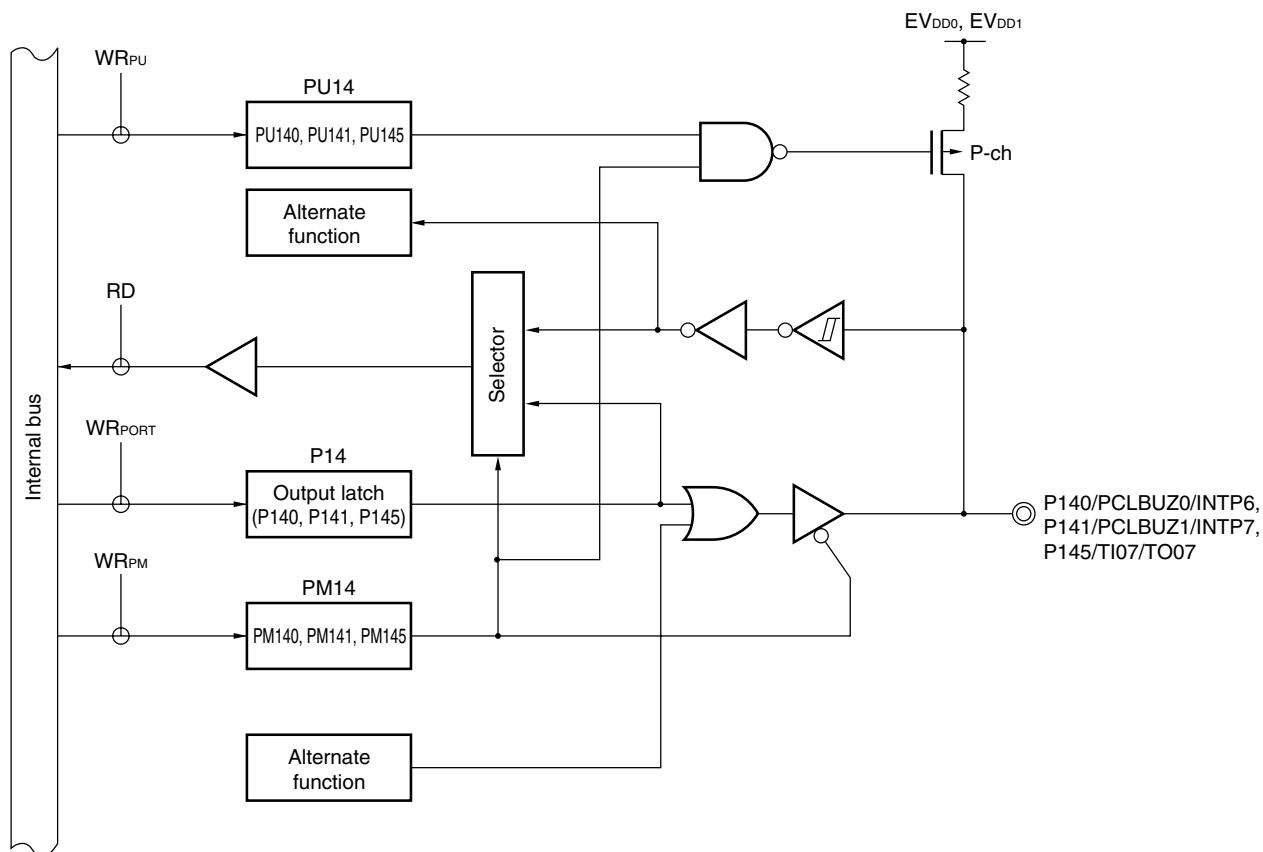
This port can also be used for timer I/O, external interrupt request input, clock/buzzer output, serial interface data I/O, and clock I/O.

Reset signal generation sets port 14 to input mode.

Figures 4-45 to 4-47 show block diagrams of port 14.

- Cautions**
1. To use P142/SCK20/SCL20, P143/SI20/RxD2/SDA20, or P144/SO20/TxD2 as a general-purpose port, note the serial array unit 1 setting. For details, refer to the following tables.
    - Table 13-9 Relationship Between Register Settings and Pins (Channel 0 of Unit 1: CSI20, UART2 Transmission, IIC20)
    - Table 13-10 Relationship Between Register Settings and Pins (Channel 1 of Unit 1: CSI21, UART2 Reception, IIC21)
  2. To use P145/TI07/TO07 as a general-purpose port, set bit 7 (TO07) of timer output register 0 (TO0) and bit 7 (TOE07) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.
  3. To use P140/PCLBUZ0/INTP6 or P141/PCLBUZ1/INTP7 as a general-purpose port, set bit 7 of clock output select registers 0 and 1 (CKS0, CKS1) to “0”, which is the same as their default status settings.

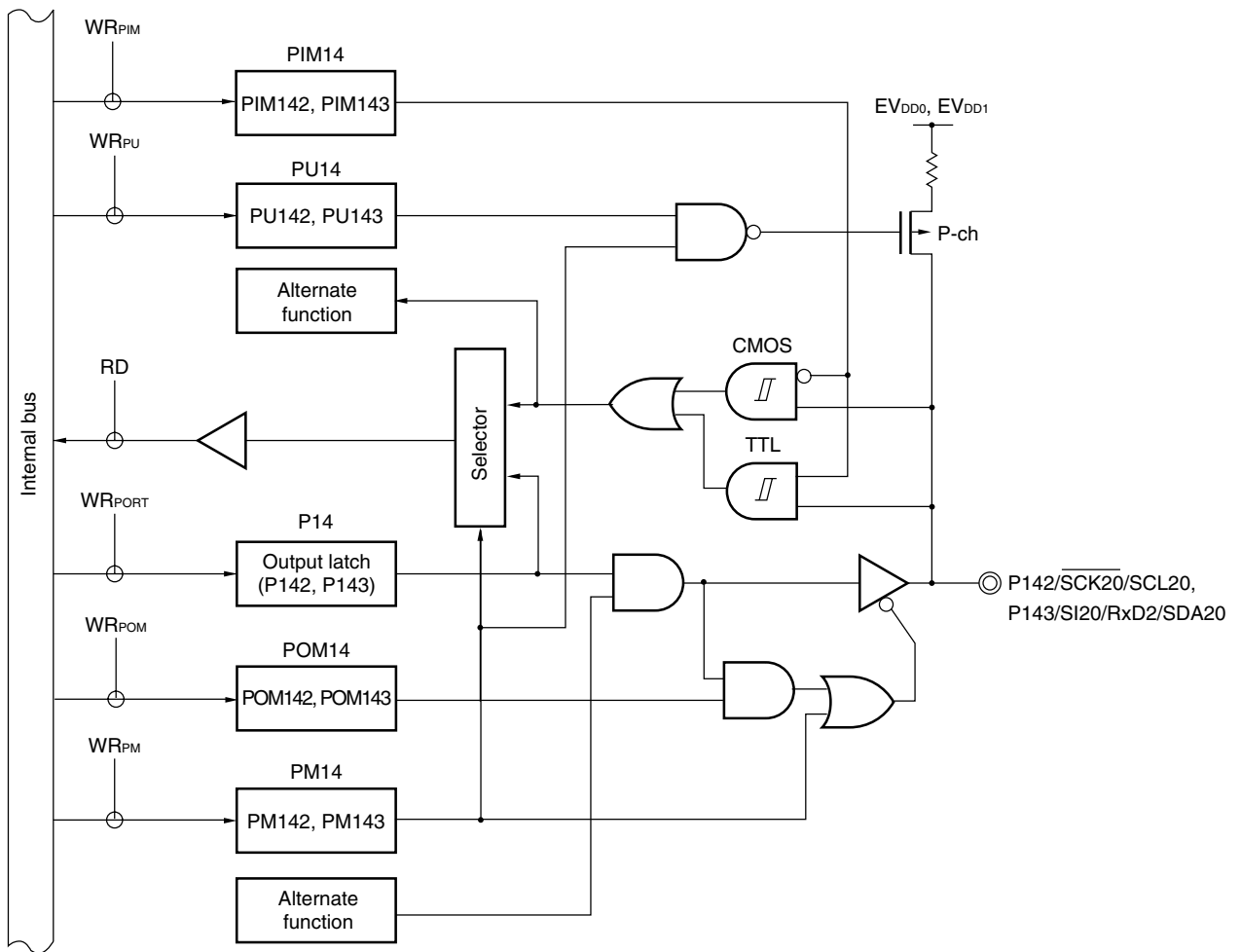
Figure 4-45. Block Diagram of P140, P141, and P145



- P14: Port register 14
- PU14: Pull-up resistor option register 14
- PM14: Port mode register 14
- RD: Read signal
- WR<sub>xx</sub>: Write signal

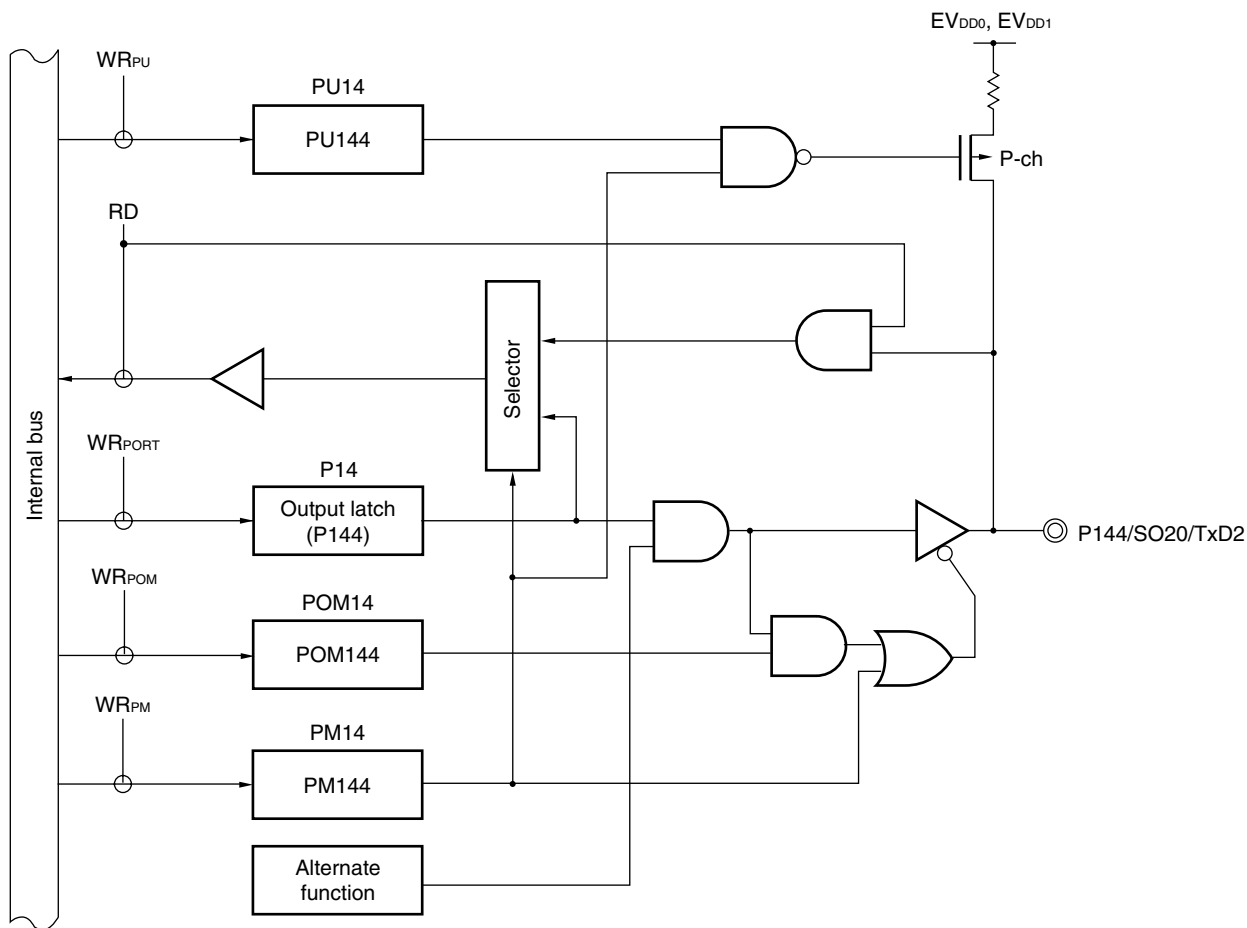


Figure 4-46. Block Diagram of P142 and P143



- P14: Port register 14
- PU14: Pull-up resistor option register 14
- PM14: Port mode register 14
- PIM14: Port input mode register 14
- POM14: Port output mode register 14
- RD: Read signal
- WR<sub>xx</sub>: Write signal

Figure 4-47. Block Diagram of P14



- P14: Port register 14
- PU14: Pull-up resistor option register 14
- PM14: Port mode register 14
- POM14: Port output mode register 14
- RD: Read signal
- WR<sub>xx</sub>: Write signal

4.2.15 Port 15

Port 15 is an 8-bit I/O port with an output latch. Port 15 can be set to the input mode or output mode in 1-bit units using port mode register 15 (PM15).

This port can also be used for A/D converter analog input.

To use P150/ANI8 to P157/ANI15 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using PM15. Use these pins starting from the lower bit.

To use P150/ANI8 to P157/ANI15 as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM15.

Table 4-5. Setting Functions of P150/ANI8 to P157/ANI15 Pins

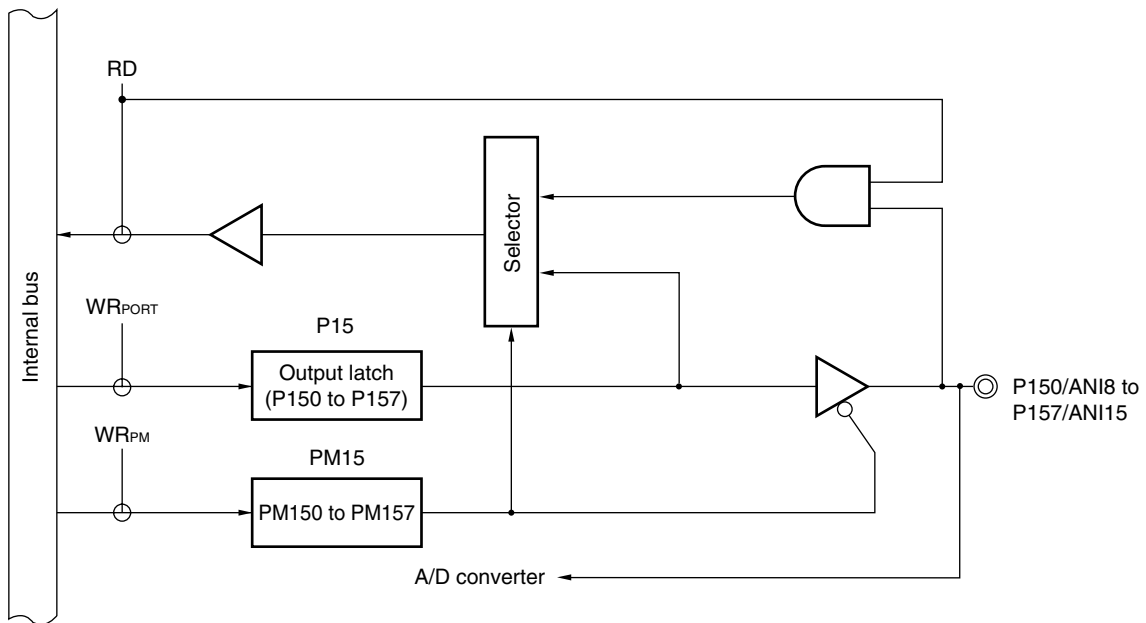
ADPC	PM15	ADS	P150/ANI8 to P157/ANI15 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	Setting prohibited

All P150/ANI8 to P157/ANI15 are set in the digital input mode when the reset signal is generated.

Figure 4-48 shows block diagram of port 15.

**Caution** See 2.2.17 AV<sub>REF0</sub> for the voltage to be applied to the AV<sub>REF0</sub> pin when using port 15 as a digital I/O.

Figure 4-48. Block Diagram of P150 to P157



- P15: Port register 15
- PM15: Port mode register 15
- RD: Read signal
- WR<sub>xx</sub>: Write signal

4.2.16 Port 16

Port 16 is a 4-bit I/O port with an output latch. Port 16 can be set to the input mode or output mode in 1-bit units using port mode register 16 (PM16). When the P160 to P163 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 16 (PU16).

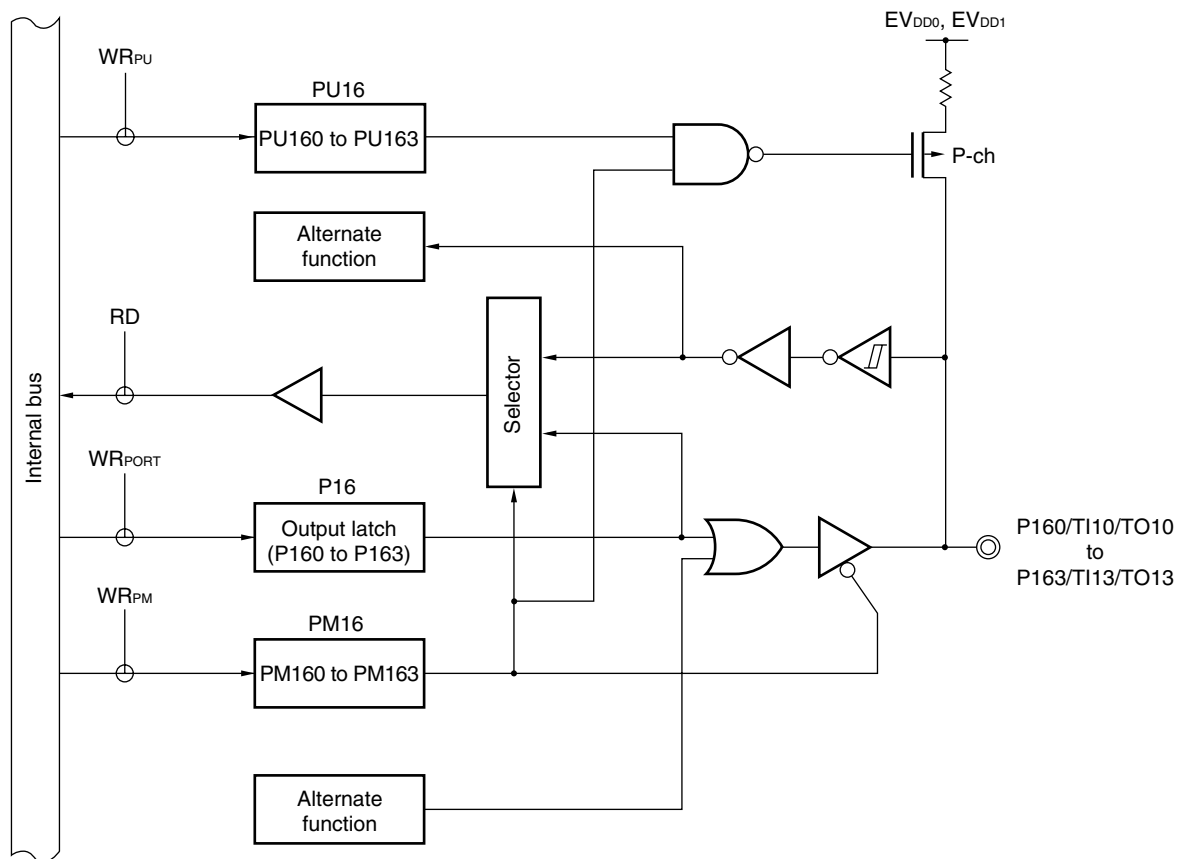
This port can also be used for timer I/O.

Reset signal generation sets port 16 to input mode.

Figure 4-49 shows a block diagram of port 16.

**Caution** To use P160/TI10/TO10 to P163/TI13/TO13 as a general-purpose port, set bits 0 to 3 (TO10 to TO13) of timer output register 0 (TO0) and bits 0 to 3 (TOE10 to TOE13) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.

Figure 4-49. Block Diagram of P160 to P163



- P16: Port register 16
- PU16: Pull-up resistor option register 16
- PM16: Port mode register 16
- RD: Read signal
- WR<sub>xx</sub>: Write signal

### 4.3 Registers Controlling Port Function

Port functions are controlled by the following six types of registers.

- Port mode registers (PM0 to PM9, PM11 to PM16)
- Port registers (P0 to P9, P11 to P16)
- Pull-up resistor option registers (PU0, PU1, PU3 to PU9, PU11 to PU14, PU16)
- Port input mode registers (PIM0, PIM4, PIM9, PIM12, PIM14)
- Port output mode registers (POM0, POM4, POM9, POM12, POM14)
- A/D port configuration register (ADPC)

#### (1) Port mode registers (PM0 to PM9, PM11 to PM16)

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 (FEH for PM13).

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

Figure 4-50. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FFF28H	FFH	R/W
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FFF29H	FFH	R/W
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110	FFF2BH	FFH	R/W
PM12	PM127	PM126	PM125	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM13	1	1	1	1	1	1	PM131	0	FFF2DH	FEH	R/W
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140	FFF2EH	FFH	R/W
PM15	PM157	PM156	PM155	PM154	PM153	PM152	PM151	PM150	FFF2FH	FFH	R/W
PM16	1	1	1	1	PM163	PM162	PM161	PM160	FFF35H	FFH	R/W
PMmn	Pmn pin I/O mode selection (m = 0 to 9, 11 to 16; n = 0 to 7)										
0	Output mode (output buffer on)										
1	Input mode (output buffer off)										

**Caution** Be sure to set bits 1 to 4 of PM12, bits 2 to 7 of PM13, bits 6 and 7 of PM14, and bits 4 to 7 of PM16 to “1”. And be sure to set bit 0 of PM13 to “0”.

**(2) Port registers (P0 to P9, P11 to P16)**

These registers write the data that is output from the chip when data is output from 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<sup>Note</sup>.

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

Reset signal generation clears these registers to 00H.

**Note** It is always 0 and never a pin level that is read out if a port is read during the input mode when P2 and P15 are set to function as an analog input for a A/D converter or P11 is set to function as an analog output for a D/A converter.

Figure 4-51. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W	
P0	P07	P06	P05	P04	P03	P02	P01	P00	FFF00H	00H (output latch)	R/W	
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W	
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W	
P3	P37	P36	P35	P34	P33	P32	P31	P30	FFF03H	00H (output latch)	R/W	
P4	P47	P46	P45	P44	P43	P42	P41	P40	FFF04H	00H (output latch)	R/W	
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFF05H	00H (output latch)	R/W	
P6	P67	P66	P65	P64	P63	P62	P61	P60	FFF06H	00H (output latch)	R/W	
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W	
P8	P87	P86	P85	P84	P83	P82	P81	P80	FFF08H	00H (output latch)	R/W	
P9	P97	P96	P95	P94	P93	P92	P91	P90	FFF09H	00H (output latch)	R/W	
P11	P117	P116	P115	P114	P113	P112	P111	P110	FFF0BH	00H (output latch)	R/W	
P12	P127	P126	P125	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W <sup>Note</sup>	
P13	0	0	0	0	0	0	P131	P130	FFF0DH	00H (output latch)	R/W	
P14	0	0	P145	P144	P143	P142	P141	P140	FFF0EH	00H (output latch)	R/W	
P15	P157	P156	P155	P154	P153	P152	P151	P150	FFF0FH	00H (output latch)	R/W	
P16	0	0	0	0	P163	P162	P161	P160	FFF34H	00H (output latch)	R/W	
Pmn	m = 0 to 9, 11 to 16; n = 0 to 7											
	Output data control (in output mode)						Input data read (in input mode)					
0	Output 0						Input low level					
1	Output 1						Input high level					

**Note** P121 to P124 are read-only.



**(3) Pull-up resistor option registers (PU0, PU1, PU3 to PU9, PU11 to PU14, PU16)**

These registers specify whether the on-chip pull-up resistors of P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P131, P140 to P145, and P160 to P163 are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode of the pins to which the use of an on-chip pull-up resistor has been specified in PU0, PU1, PU3 to PU9, PU11 to PU14, PU16. 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 PU0, PU1, PU3 to PU9, PU11 to PU14, PU16.

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

Reset signal generation clears these registers to 00H.

Figure 4-52. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30	F0033H	00H	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	F0034H	00H	R/W
PU5 <sup>Note</sup>	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU6	PU67	PU66	PU65	PU64	0	0	0	0	F0036H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU8 <sup>Note</sup>	PU87	PU86	PU85	PU84	PU83	PU82	PU81	PU80	F0038H	00H	R/W
PU9	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90	F0039H	00H	R/W
PU11	PU117	PU116	PU115	PU114	PU113	PU112	0	0	F003BH	00H	R/W
PU12	PU127	PU126	PU125	0	0	0	0	PU120	F003CH	00H	R/W
PU13	0	0	0	0	0	0	PU131	0	F003DH	00H	R/W
PU14	0	0	PU145	PU144	PU143	PU142	PU141	PU140	F003EH	00H	R/W
PU16	0	0	0	0	PU163	PU162	PU161	PU160	F0064H	00H	R/W
PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 9, 11 to 14, 16; n = 0 to 7)										
0	On-chip pull-up resistor not connected										
1	On-chip pull-up resistor connected										

**Note** When the external expansion function is used, On-chip pull-up resistors cannot be connected, regardless of the settings of PU5 and PU8 registers.

**(4) Port input mode registers (PIM0, PIM4, PIM9, PIM12, PIM14)**

These registers set the input buffer of P03, P04, P43, P44, P95, P96, P125, P126, P142, and P143 in 1-bit units. TTL input buffer can be selected during serial communication with an external device of the different potential. These registers can be set by a 1-bit or 8-bit memory manipulation instruction. Reset signal generation clears these registers to 00H.

**Figure 4-53. Format of Port Input Mode Register**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	PIM04	PIM03	0	0	0	F0040H	00H	R/W
PIM4	0	0	0	PIM44	PIM43	0	0	0	F0044H	00H	R/W
PIM9	0	PIM96	PIM95	0	0	0	0	0	F0049H	00H	R/W
PIM12	0	PIM126	PIM125	0	0	0	0	0	F004CH	00H	R/W
PIM14	0	0	0	0	PIM143	PIM142	0	0	F004EH	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0, 4, 9, 12, 14; n = 2 to 6)
0	Normal input buffer
1	TTL input buffer

**(5) Port output mode registers (POM0, POM4, POM9, POM12, POM14)**

These registers set the output mode of P02 to P04, P43, P45, P95 to P97, P125 to P127, or P142 to P144 in 1-bit units.

N-ch open drain output ( $V_{DD}$  tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDA10, SDA11, SDA20, and SDA21 pins during simplified I<sup>2</sup>C communication with an external device of the same potential.

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

Reset signal generation clears these registers to 00H.

**Figure 4-54. Format of Port Input Mode Register**

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	POM04	POM03	POM02	0	0	F0050H	00H	R/W
POM4	0	0	POM45	0	POM43	0	0	0	F0054H	00H	R/W
POM9	POM97	POM96	POM95	0	0	0	0	0	F0059H	00H	R/W
POM12	POM127	POM126	POM125	0	0	0	0	0	F005CH	00H	R/W
POM14	0	0	0	POM144	POM143	POM142	0	0	F005EH	00H	R/W
POMmn	Pmn pin output mode selection (m = 0, 4, 9, 12, 14; n = 2 to 7)										
0	Normal output mode										
1	N-ch open-drain output ( $V_{DD}$ tolerance) mode										

**(6) A/D port configuration register (ADPC)**

This register switches the P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 pins to digital I/O of port or analog input of A/D converter.

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

Reset signal generation sets this register to 10H.

**Figure 4-55. Format of A/D Port Configuration Register (ADPC)**

Address: F0017H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADP C4	ADP C3	ADP C2	ADP C1	ADP C0	Analog input (A)/digital I/O (D) switching															
					Port 15								Port 2							
					ANI15/ P157	ANI14/ P156	ANI13/ P155	ANI12/ P154	ANI11/ P153	ANI10/ P152	ANI9/ P151	ANI8/ P150	ANI7/ P27	ANI6/ P26	ANI5/ P25	ANI4/ P24	ANI3/ P23	ANI2/ P22	ANI1/ P21	ANI0/ P20
0	0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
0	0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	A	A	A	D	
0	0	0	1	0	A	A	A	A	A	A	A	A	A	A	A	A	A	D	D	
0	0	0	1	1	A	A	A	A	A	A	A	A	A	A	A	A	D	D	D	
0	0	1	0	0	A	A	A	A	A	A	A	A	A	A	A	D	D	D	D	
0	0	1	0	1	A	A	A	A	A	A	A	A	A	A	D	D	D	D	D	
0	0	1	1	0	A	A	A	A	A	A	A	A	A	D	D	D	D	D	D	
0	0	1	1	1	A	A	A	A	A	A	A	A	A	D	D	D	D	D	D	
0	1	0	0	0	A	A	A	A	A	A	A	A	D	D	D	D	D	D	D	
0	1	0	0	1	A	A	A	A	A	A	A	D	D	D	D	D	D	D	D	
0	1	0	1	0	A	A	A	A	A	A	D	D	D	D	D	D	D	D	D	
0	1	0	1	1	A	A	A	A	A	D	D	D	D	D	D	D	D	D	D	
0	1	1	0	0	A	A	A	D	D	D	D	D	D	D	D	D	D	D	D	
0	1	1	0	1	A	A	A	D	D	D	D	D	D	D	D	D	D	D	D	
0	1	1	1	0	A	A	D	D	D	D	D	D	D	D	D	D	D	D	D	
0	1	1	1	1	A	D	D	D	D	D	D	D	D	D	D	D	D	D	D	
1	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	
Other than above					Setting prohibited															

- Cautions**
1. Set the channel used for A/D conversion to the input mode by using port mode registers 2 and 15 (PM2, PM15).
  2. Do not set the pin set by ADPC as digital I/O by analog input channel specification register (ADS).
  3. P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 are set as analog inputs in the order of P157/ANI15, ..., P150/ANI8, P27/ANI7, ..., P20/ANI0 by the A/D port configuration register (ADPC). When using P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 as analog inputs, start designing from P157/ANI15.

## 4.4 Port Function Operations

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

### 4.4.1 Writing to I/O port

#### (1) Output mode

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

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

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

#### (2) Input mode

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

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

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

### 4.4.2 Reading from I/O port

#### (1) Output mode

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

#### (2) Input mode

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

### 4.4.3 Operations on I/O port

#### (1) Output mode

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

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

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

#### (2) Input mode

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

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

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

When parts of ports 0, 4, 9, 12 and 14 operate with  $V_{DD} = 4.0\text{ V}$  to  $5.5\text{ V}$ , I/O connections with an external device that operates on 2.5 V or 3 V power supply voltage are possible.

Regarding inputs, CMOS/TTL switching is possible on a bit-by-bit basis by port input mode registers (PIM0, PIM4, PIM9, PIM12, PIM14).

Moreover, regarding outputs, different potentials can be supported by switching the output buffer to the N-ch open drain ( $V_{DD}$  withstand voltage) by the port output mode registers (POM0, POM4, POM9, POM12, POM14).

##### (1) Setting procedure when using I/O pins of UART1, UART2, CSI01, CSI10, CSI11, CSI20 and CSI21 functions

###### (a) Use as 2.5 V or 3 V input port

<1> After reset release, the port mode is the input mode (Hi-Z).

<2> If pull-up is needed, externally pull up the pin to be used (on-chip pull-up resistor cannot be used).

In case of UART1:	P03
In case of UART2:	P143
In case of CSI01:	P43, P44
In case of CSI10:	P03, P04
In case of CSI11:	P95, P96
In case of CSI20:	P142, P143
In case of CSI21:	P125, P126

<3> Set the corresponding bit of the PIMn register to 1 to switch to the TTL input buffer.

<4>  $V_{IH}/V_{IL}$  operates on 2.5 V or 3 V operating voltage.

###### (b) Use as 2.5 V or 3 V output port

<1> After reset release, the port mode changes to the input mode (Hi-Z).

<2> Pull up externally the pin to be used (on-chip pull-up resistor cannot be used).

In case of UART1:	P02
In case of UART2:	P144
In case of CSI01:	P43, P45
In case of CSI10:	P02, P04
In case of CSI11:	P95, P97
In case of CSI20:	P142, P144
In case of CSI21:	P125, P127

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

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

<5> Set the output mode by manipulating the PMn register.

At this time, the output data is high level, so the pin is in the Hi-Z state.

<6> Operation is done only in the low level according to the operating status of the serial array unit.

**Remark** n = 0, 4, 9, 12, 14

**(2) Setting procedure when using I/O pins of simplified IIC10, IIC11, IIC20 and IIC21 functions**

- <1> After reset release, the port mode is the input mode (Hi-Z).
- <2> Externally pull up the pin to be used (on-chip pull-up resistor cannot be used).

In case of simplified IIC10: P03, P04  
In case of simplified IIC11: P95, P96  
In case of simplified IIC20: P142, P143  
In case of simplified IIC21: P125, P126

- <3> Set the output latch of the corresponding port to 1.
- <4> Set the corresponding bit of the POMn register to 1 to set the N-ch open drain output ( $V_{DD}$  withstand voltage) mode.
- <5> Set the corresponding bit of the PMn register to the output mode (data I/O is possible in the output mode).  
At this time, the output data is high level, so the pin is in the Hi-Z state.
- <6> Enable the operation of the serial array unit and set the mode to the simplified I<sup>2</sup>C mode.

**Remark** n = 0, 9, 12, 14



#### 4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function

To use the alternate function of a port pin, set the port mode register and output latch as shown in Table 4-6.

**Table 4-6. Settings of Port Mode Register and Output Latch When Using Alternate Function (1/3)**

Pin Name	Alternate Function		PM <sub>xx</sub>	P <sub>xx</sub>
	Function Name	I/O		
P00	TI00	Input	1	×
P01	TO00	Output	0	0
P02	SO10	Output	0	1
	TxD1	Output	0	1
P03	SI10	Input	1	×
	RxD1	Input	1	×
	SDA10	I/O	0	1
P04	SCK10	Input	1	×
		Output	0	1
	SCL10	I/O	0	1
P05	CLKOUT	Output	0	0
P06	WAIT	Input	1	×
P10	SCK00	Input	1	×
		Output	0	1
	EX24	Output	0	0
P11	SI00	Input	1	×
	RxD0	Input	1	×
	EX25	Output	0	0
P12	SO00	Output	0	1
	TxD0	Output	0	1
	EX26	Output	0	0
P13	TxD3	Output	0	1
	EX27	Output	0	0
P14	RxD3	Input	1	×
	EX28	Output	0	0
P15	RTCDIV	Output	0	0
	RTCCL	Output	0	0
	EX29	Output	0	0
P16	TI01	Input	1	×
	TO01	Output	0	0
	INTP5	Input	1	×
	EX30	Output	0	0

**Remark** ×: don't care

PM<sub>xx</sub>: Port mode register

P<sub>xx</sub>: Port output latch

Table 4-6. Settings of Port Mode Register and Output Latch When Using Alternate Function (2/3)

Pin Name	Alternate Function		PM <sub>xx</sub>	P <sub>xx</sub>
	Function Name	I/O		
P17	TI02	Input	1	×
	TO02	Output	0	0
	EX31	Output	0	0
P20 to P27 <sup>Note 1</sup>	ANI0 to ANI7 <sup>Note 1</sup>	Input	1	×
P30	RTC1HZ	Output	0	0
	INTP3	Input	1	×
P31	TI03	Input	1	×
	TO03	Output	0	0
	INTP4	Input	1	×
P40	TOOL0	I/O	×	×
P41	TOOL1	Output	×	×
P42	TI04	Input	1	×
	TO04	Output	0	0
P43	SCK01	Input	1	×
		Output	0	1
P44	SI01	Input	1	×
P45	SO01	Output	0	1
P46	TI05	Input	1	×
	TO05	Output	0	0
	INTP1	Input	1	×
P47	INTP2	Input	1	×
P50 to P57	EX8 to EX15 <sup>Note 2</sup>	I/O	× <sup>Note 4</sup>	
P60	SCL0	I/O	0	0
P61	SDA0	I/O	0	0
P64	$\overline{RD}$	Output	0	0
P65	$\overline{WR0}$	Output	0	0
P66	$\overline{WR1}$	Output	0	0
P67	ASTB	Output	0	0
P70 to P73	KR0 to KR3	Input	1	×
	EX16 to EX19	Output	0	0
P74 to P77	INTP8 to INTP11	Input	1	×
	KR4 to KR7	Input	1	×
	EX20 to EX23	Output	0	0
P80 to P87	EX0 to EX7 <sup>Note 3</sup>	I/O	× <sup>Note 4</sup>	
P90 to P93	EX32 to EX35	Output	0	0
P95	SCK11	Input	1	×
		Output	0	1
	SCL11	I/O	0	1

**Remark** ×: don't care  
 PM<sub>xx</sub>: Port mode register  
 P<sub>xx</sub>: Port output latch

(Notes 1 to 4 are listed on the next page after next.)

Table 4-6. Settings of Port Mode Register and Output Latch When Using Alternate Function (3/3)

Pin Name	Alternate Function		PM <sub>xx</sub>	P <sub>xx</sub>
	Function Name	I/O		
P96	SI11	Input	1	×
	SDA1	I/O	0	1
P97	SO11	Output	0	1
P110, P111	ANO0, ANO1 <sup>Note 5</sup>	Output	1	×
P120	INTP0	Input	1	×
	EXLVI	Input	1	×
P125	SCK21	Input	1	×
		Output	0	1
	SCL21	I/O	0	1
P126	SI21	Input	1	×
	SDA21	I/O	0	1
P127	SO21	Output	0	1
P131	TI06	Input	1	×
	TO06	Output	0	0
P140	PCLBUZ0	Output	0	0
	INTP6	Input	1	×
P141	PCLBUZ1	Output	0	0
	INTP7	Input	1	×
P142	SCK20	Input	1	×
		Output	0	1
	SCL20	I/O	0	1
P143	SI20	Input	1	×
	RxD2	Input	1	×
	SDA20	I/O	0	1
P144	SO20	Output	0	1
	TxD2	Output	0	1
P145	TI07	Input	1	×
	TO07	Output	0	0
P150 to P157 <sup>Note 1</sup>	ANI8 to ANI15 <sup>Note 1</sup>	Input	1	×
P160 to P163	TI10 to TI13	Input	1	×
	TO10 to TO13	Output	0	0

**Remark** ×: don't care  
 PM<sub>xx</sub>: Port mode register  
 P<sub>xx</sub>: Port output latch

(Notes 1 and 5 are listed on the next page)

- Notes 1.** The functions of the ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157 pins can be selected by using the A/D port configuration register (ADPC), the analog input channel specification register (ADS), PM2, and PM15.

**Table 4-7. Setting Functions of ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157 Pins**

ADPC	PM2, PM15	ADS	ANI0/P20 to ANI7/P27, ANI8/P150 to ANI15/P157 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

2. The functions of EX8 to EX15 are selected depending on the setting of the memory extension mode control register (MEM). For details, see **CHAPTER 5 EXTERNAL BUS INTERFACE**.
3. When the external bus interface is enabled to operate (EXEN = 1), the functions of EX0 to EX7 are automatically selected.
4. When using an alternate function of these pins, select the function by using the memory extension mode control register (MEM).
5. When the D/A converter operation is enabled (DACE<sub>n</sub> = 1), the function of ANO<sub>n</sub> is automatically selected. However, set port mode register 11 in the input mode (PM11<sub>n</sub> = 1).

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

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

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

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

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

A 1-bit manipulation instruction is executed in the following order in the 78K0R/KH3.

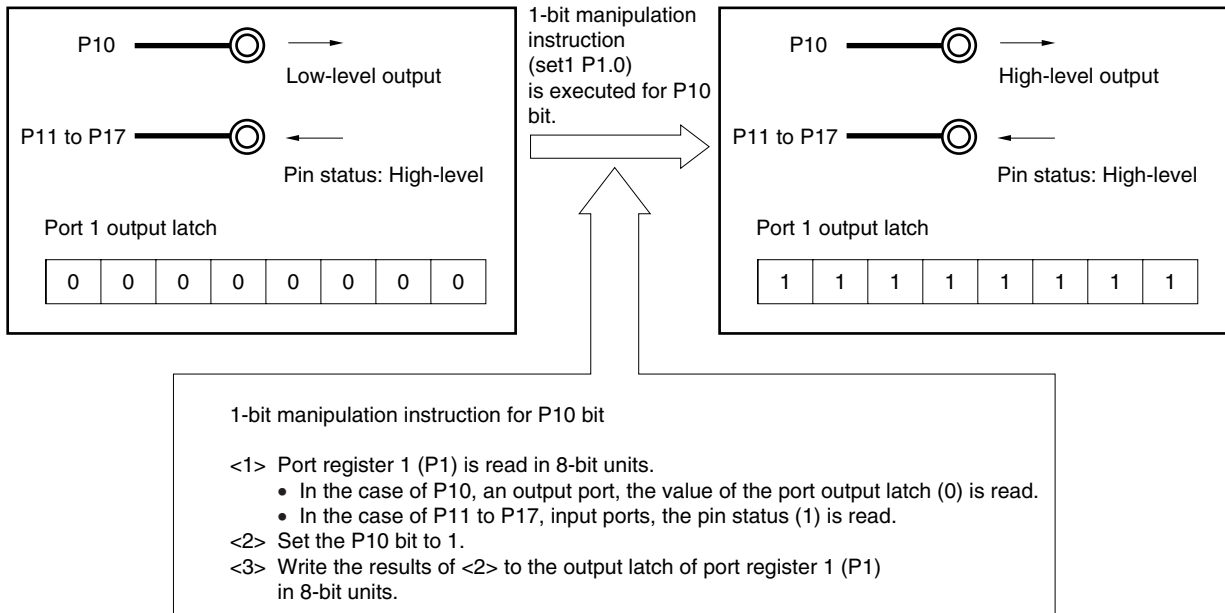
- <1> The Pn register is read in 8-bit units.
- <2> The targeted one bit is manipulated.
- <3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P17, which are input ports, are read. If the pin statuses of P11 to P17 are high level at this time, the read value is FEH.

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

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

Figure 4-56. Bit Manipulation Instruction (P10)



## CHAPTER 5 EXTERNAL BUS INTERFACE

### 5.1 Functions of External Bus Interface

The external bus interface function is used to connect an external device to an area other than the internal ROM, RAM, and SFR areas. An external device is connected by using ports 0, 1, and 5 to 9. Ports 0, 1, and 5 to 9 control signals such as address/data, read/write strobe, wait, and address strobe.

The external bus interface has the following features.

- The number of address bits can be selected from 8, 12, 16, and 20.
- Data bus supporting 8 bits and 16 bits
- Multiplexed bus and separate bus are supported.
- Separate bus mode and 16-bit bus mode are selected when an instruction is fetched from external memory.

The following table shows the pin functions in an external memory extension mode.

Pin Function When External Device Is Connected		Alternate-Function Pin
Name	Function	
EX0 to EX7	External extension I/O (multiplexed address/data bus, data bus)	P80 to P87
EX8 to EX15	External extension I/O (multiplexed address/data bus, address bus, data bus)	P50 to P57
EX16 to EX23	External extension output (address bus)	P70/KR0 to P77/KR7/INTP11
EX24 to EX31	External extension output (address bus)	P10/ $\overline{\text{SCK00}}$ to P17/TI02/TO02
EX32 to EX35	External extension output (address bus)	P90 to P93
$\overline{\text{RD}}$	Read strobe signal	P64
$\overline{\text{WR0}}$	Write strobe signal (8-bit bus mode, 16-bit bus mode (lower byte))	P65
$\overline{\text{WR1}}$	Write strobe signal (16-bit bus mode (higher byte))	P66
CLKOUT	Internal system clock output	P05
$\overline{\text{WAIT}}$	Wait signal	P06
ASTB	Address strobe signal	P67

The function of the external bus interface pins differs depending on the set mode.

External Extension Mode		Pin	EX35 to EX32	EX31 to EX28	EX27 to EX24	EX23 to EX20	EX19 to EX16	EX15 to EX12	EX11 to EX8	EX7 to EX0
		Multiplexed bus mode								
8-bit bus mode	256-byte extension mode	-	-	-	-	-	-	-	-	AD7 to AD0
	4 KB extension mode	-	-	-	-	-	-	-	A11 to A8	AD7 to AD0
	64 KB extension mode	-	-	-	-	-	-	A15 to A12	A11 to A8	AD7 to AD0
	Full address mode	-	-	-	-	-	A19 to A16	A15 to A12	A11 to A8	AD7 to AD0
16-bit bus mode	256-byte extension mode	-	-	-	-	-	-	D15 to D12	D11 to D8	AD7 to AD0
	4 KB extension mode	-	-	-	-	-	-	D15 to D12	AD11 to AD8	AD7 to AD0
	64 KB extension mode	-	-	-	-	-	-	AD15 to AD12	AD11 to AD8	AD7 to AD0
	Full address mode	-	-	-	-	-	A19 to A16	AD15 to AD12	AD11 to AD8	AD7 to AD0
Separate bus mode	8-bit bus mode	256-byte extension mode	-	-	-	-	-	A7 to A4	A3 to A0	D7 to D0
		4 KB extension mode	-	-	-	-	A11 to A8	A7 to A4	A3 to A0	D7 to D0
		64 KB extension mode	-	-	-	A15 to A12	A11 to A8	A7 to A4	A3 to A0	D7 to D0
		Full address mode	-	-	A19 to A16	A15 to A12	A11 to A8	A7 to A4	A3 to A0	D7 to D0
	16-bit bus mode	256-byte extension mode	-	-	-	A7 to A4	A3 to A0	D15 to D12	D11 to D8	D7 to D0
		4 KB extension mode	-	-	A11 to A8	A7 to A4	A3 to A0	D15 to D12	D11 to D8	D7 to D0
		64 KB extension mode	-	A15 to A12	A11 to A8	A7 to A4	A3 to A0	D15 to D12	D11 to D8	D7 to D0
		Full address mode	A19 to A16	A15 to A12	A11 to A8	A7 to A4	A3 to A0	D15 to D12	D11 to D8	D7 to D0

**Remark** EXxx: Pin name  
 Axx: Address bus  
 Dxx: Data bus  
 ADxx: Multiplexed address/data bus  
 -: External bus interface is not used. These pins can be used as port pins.

The memory maps when using the external bus interface function are as follows.

Figure 5-1. Memory Map When Using External Bus Interface Function (1/3)

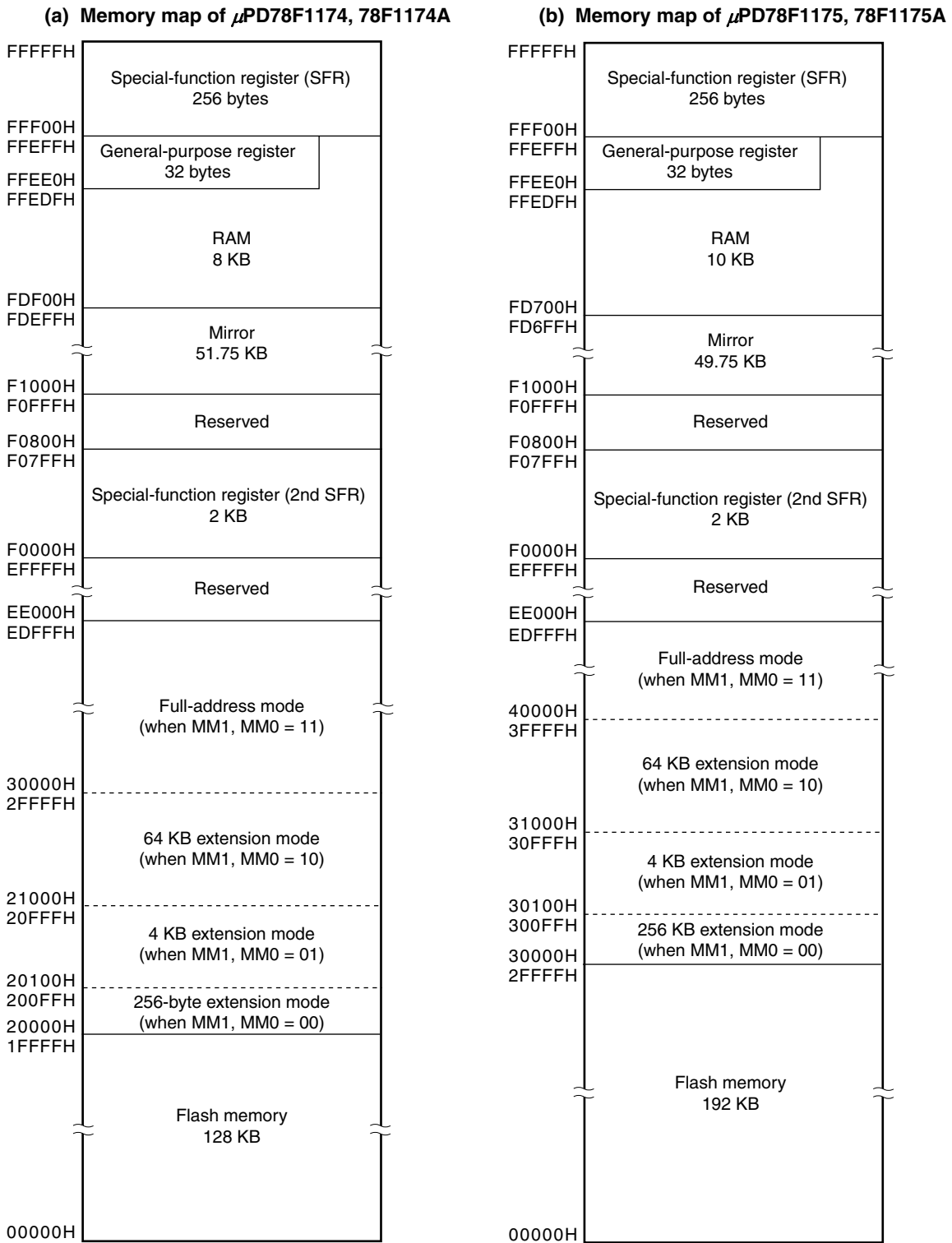




Figure 5-1. Memory Map When Using External Bus Interface Function (2/3)

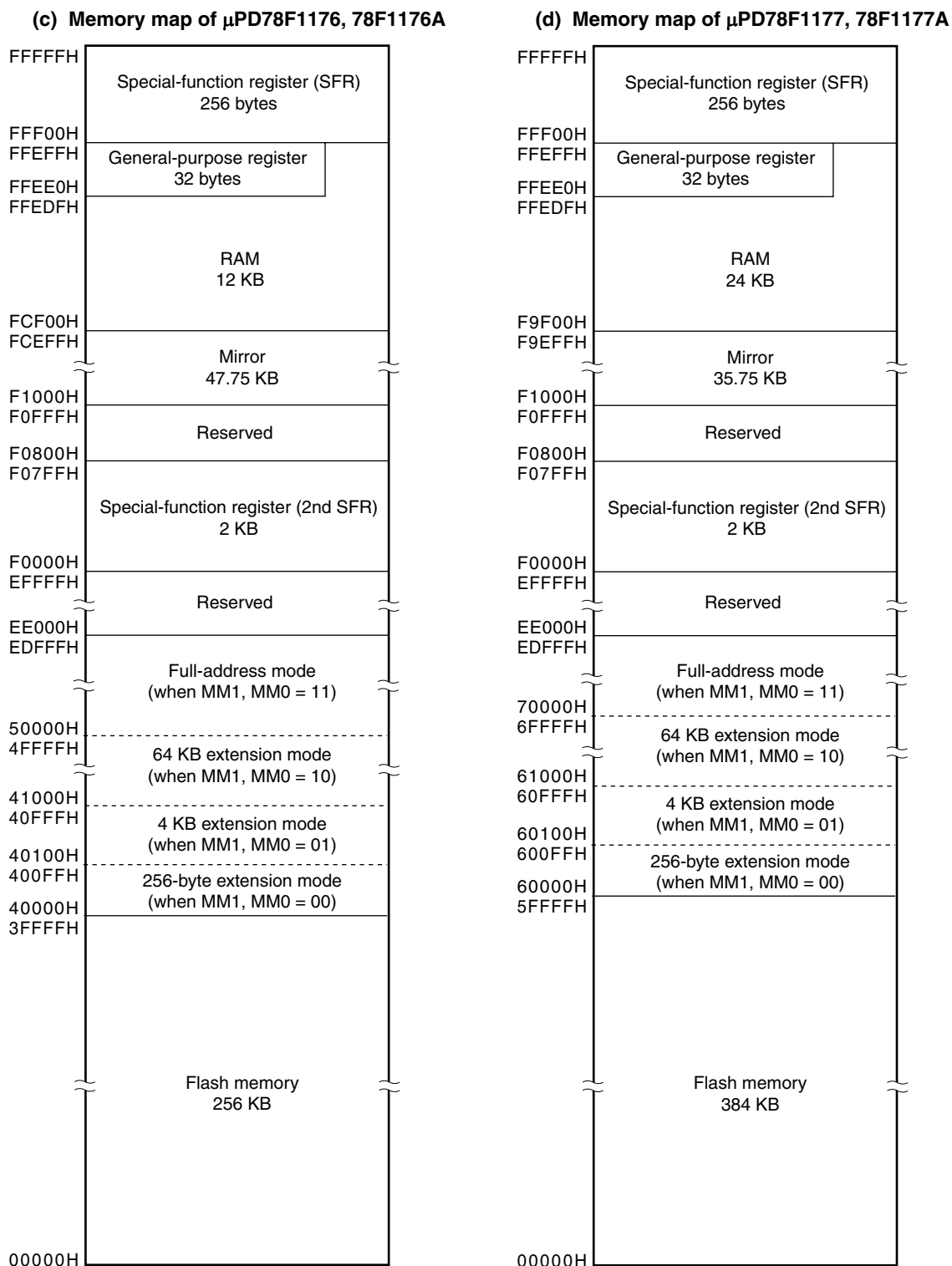
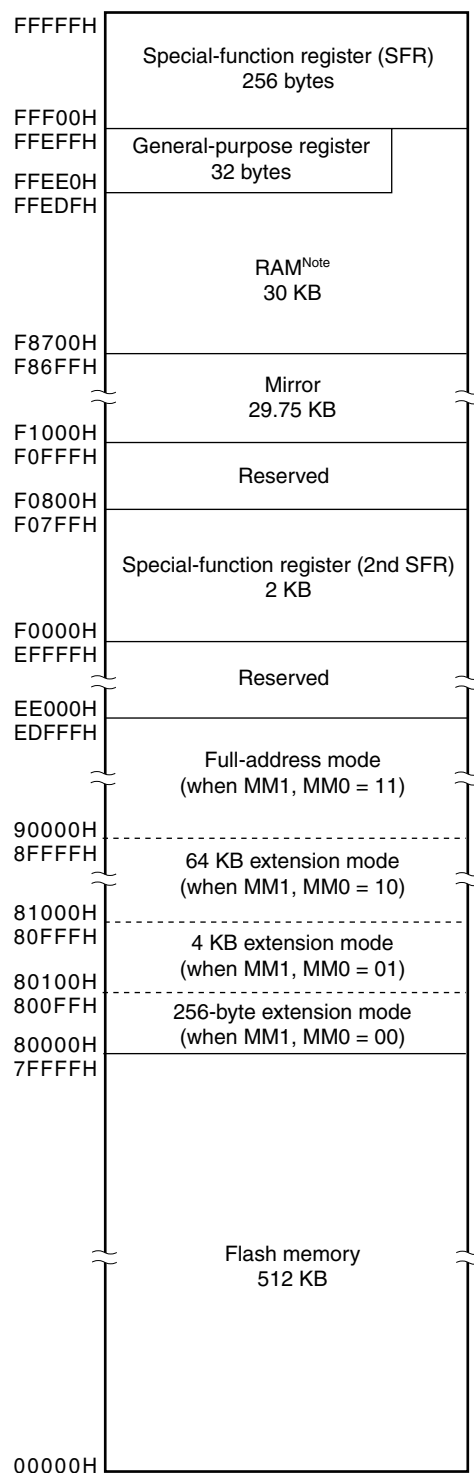


Figure 5-1. Memory Map When Using External Bus Interface Function (3/3)

(e) Memory map of  $\mu$ PD78F1178, 78F1178A

**Note** Use of the area F8700H to F86FFH is prohibited when using the self-programming function. Since this area is used for self-programming library.

## 5.2 Registers Controlling External Bus Interface Functions

The external bus interface function is controlled by the following two registers.

- Peripheral enable register 1 (PER1)
- Memory extension mode control register (MEM)
- Port mode registers 0, 1, 5, 6, 7, 8, 9 (PM0, PM1, PM5, PM6, PM7, PM8, PM9)
- Port registers 0, 1, 5, 6, 7, 8, 9 (P0, P1, P5, P6, P7, P8, P9)

### (1) Peripheral enable register 1 (PER1)

PER1 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the external bus interface is used, be sure to set bit 0 (EXBEN) of this register to 1.

PER1 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 1 (PER1)**

Address: F00F1H After reset: 00H R/W

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

EXBEN	Control of external bus interface input clock
0	Stops supply of input clock. <ul style="list-style-type: none"> <li>• SFR used by external bus interface cannot be written.</li> <li>• External bus interface is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by external bus interface can be read/written.</li> </ul>

**Caution** When setting the external bus interface, be sure to set EXBEN to 1 first. If EXBEN = 0, writing to a control register of the external bus interface is ignored, and, even if the register is read, only the default value is read (except for port mode registers 0, 1, 5, 6, 7, 8, 9 (PM0, PM1, PM5, PM6, PM7, PM8, PM9) and port registers 0, 1, 5, 6, 7, 8, 9 (P0, P1, P5, P6, P7, P8, P9)).

**(2) Memory extension mode control register (MEM)**

MEM is a register that sets an external extension area.

MEM can be set by a 1-bit or 8-bit manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 5-3. Format of Memory Extension Mode Control Register (MEM)**

Address: FFFFFH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MEM	EXEN	EXWEN	EW1	EW0	MM3	MM2	MM1	MM0

EXEN	External bus interface enable flag
0	Single-chip mode (Port function is valid.)
1	External bus interface is used.

EXWEN	External wait pin enable signal
0	External wait pin is not used and can be used as a port.
1	External wait pin is used.

EW1	EW0	CLKOUT pin output clock selection
0	0	$f_{CLK}$
0	1	$f_{CLK}/2$
1	0	$f_{CLK}/3$
1	1	$f_{CLK}/4$

MM3	Bus mode switching of external bus interface
0	Multiplexed bus mode (Can be selected only for accessing memory and cannot be selected for fetching.)
1	Separate bus mode

MM2	Bus width selection of external bus interface
0	8-bit bus mode <sup>Note</sup> (Can be selected only for accessing memory and cannot be selected for fetching.)
1	16-bit bus mode

MM1	MM0	Mode selection
0	0	256-byte extension mode. 8 address bus pins are used.
0	1	4 KB extension mode. 12 address bus pins are used.
1	0	64 KB extension mode. 16 address bus pins are used.
1	1	Full address mode. 20 address bus pins are used.

**Note** In the 8-bit bus mode, 16-bit access instructions cannot be used.

The function of the external bus interface pins differs depending on the setting of the memory extension mode control register (MEM).

MM3	MM2	MM1	MM0	EX35 to EX32	EX31 to EX28	EX27 to EX24	EX23 to EX20	EX19 to EX16	EX15 to EX12	EX11 to EX8	EX7 to EX0
0	0	0	0	–	–	–	–	–	–	–	AD7 to AD0
0	0	0	1	–	–	–	–	–	–	A11 to A8	AD7 to AD0
0	0	1	0	–	–	–	–	–	A15 to A12	A11 to A8	AD7 to AD0
0	0	1	1	–	–	–	–	A19 to A16	A15 to A12	A11 to A8	AD7 to AD0
0	1	0	0	–	–	–	–	–	D15 to D12	D11 to D8	AD7 to AD0
0	1	0	1	–	–	–	–	–	D15 to D12	AD11 to AD8	AD7 to AD0
0	1	1	0	–	–	–	–	–	AD15 to AD12	AD11 to AD8	AD7 to AD0
0	1	1	1	–	–	–	–	A19 to A16	AD15 to AD12	AD11 to AD8	AD7 to AD0
1	0	0	0	–	–	–	–	–	A7 to A4	A3 to A0	D7 to D0
1	0	0	1	–	–	–	–	A11 to A8	A7 to A4	A3 to A0	D7 to D0
1	0	1	0	–	–	–	A15 to A12	A11 to A8	A7 to A4	A3 to A0	D7 to D0
1	0	1	1	–	–	A19 to A16	A15 to A12	A11 to A8	A7 to A4	A3 to A0	D7 to D0
1	1	0	0	–	–	–	A7 to A4	A3 to A0	D15 to D12	D11 to D8	D7 to D0
1	1	0	1	–	–	A11 to A8	A7 to A4	A3 to A0	D15 to D12	D11 to D8	D7 to D0
1	1	1	0	–	A15 to A12	A11 to A8	A7 to A4	A3 to A0	D15 to D12	D11 to D8	D7 to D0
1	1	1	1	A19 to A16	A15 to A12	A11 to A8	A7 to A4	A3 to A0	D15 to D12	D11 to D8	D7 to D0

EXEN	EXWEN	MM3	MM2	CLKOUT	ASTB	$\overline{RD}$	$\overline{WR0}$	$\overline{WR1}$	$\overline{WAIT}$
0	X	X	X	–	–	–	–	–	–
1	0	0	0	CLKOUT	ASTB	RD	Write strobe	–	–
1	0	0	1	CLKOUT	ASTB	RD	Low bytes write strobe	High bytes write strobe	–
1	0	1	0	CLKOUT	–	RD	Write strobe	–	–
1	0	1	1	CLKOUT	–	RD	Low bytes write strobe	High bytes write strobe	–
1	1	0	0	CLKOUT	ASTB	RD	Write strobe	–	WAIT
1	1	0	1	CLKOUT	ASTB	RD	Low bytes write strobe	High bytes write strobe	WAIT
1	1	1	0	CLKOUT	–	RD	Write strobe	–	WAIT
1	1	1	1	CLKOUT	–	RD	Low bytes write strobe	High bytes write strobe	WAIT

**Remark** EXxx: Pin name

Axx: Address bus

Dxx: Data bus

ADxx: Multiplexed address/data bus

–: External bus interface is not used. These pins can be used as port pins.

### 5.3 Setting Port Mode Register and Output Latch

Set the port mode register and output latch as follows when using the external bus interface.

EXEN	MM3	MM2	MM1	MM0	P93/EX35 to P90/EX32		P17/EX31 to P14/EX28		P13/EX27 to P10/EX24		P77/EX23 to P74/EX20		P73/EX19 to P70/EX16		P57/EX15 to P54/EX12		P53/EX11 to P50/EX8		P87/EX7 to P80/EX0	
					PM9x	P9x	PM1x	P1x	PM1x	P1x	PM7x	P7x	PM7x	P7x	PM5x	P5x	PM5x	P5x	PM8x	P8x
					0	X	X	X	X	-	-	-	-	-	-	-	-	-	-	-
1	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X
1	0	0	0	1	-	-	-	-	-	-	-	-	-	-	-	X	X	X	X	X
1	0	0	1	0	-	-	-	-	-	-	-	-	-	X	X	X	X	X	X	X
1	0	0	1	1	-	-	-	-	-	-	-	0	0	X	X	X	X	X	X	X
1	0	1	0	0	-	-	-	-	-	-	-	-	-	X	X	X	X	X	X	X
1	0	1	0	1	-	-	-	-	-	-	-	-	-	X	X	X	X	X	X	X
1	0	1	1	0	-	-	-	-	-	-	-	-	-	X	X	X	X	X	X	X
1	0	1	1	1	-	-	-	-	-	-	-	0	0	X	X	X	X	X	X	X
1	1	0	0	0	-	-	-	-	-	-	-	-	-	X	X	X	X	X	X	X
1	1	0	0	1	-	-	-	-	-	-	-	0	0	X	X	X	X	X	X	X
1	1	0	1	1	-	-	-	-	0	0	0	0	0	X	X	X	X	X	X	X
1	1	1	0	0	-	-	-	-	-	-	0	0	0	X	X	X	X	X	X	X
1	1	1	0	1	-	-	-	-	0	0	0	0	0	X	X	X	X	X	X	X
1	1	1	1	0	-	-	0	0	0	0	0	0	0	X	X	X	X	X	X	X
1	1	1	1	1	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X

EXEN	EXWEN	MM3	P05/CLKOUT		P67/ASTB		P64/ $\overline{RD}$		P65/ $\overline{WR0}$		P66/ $\overline{WR1}$		P06/ $\overline{WAIT}$	
			PM05	P05	PM67	P67	PM64	P64	PM65	P65	PM66	P66	PM06	P06
0	X	X	-	-	-	-	-	-	-	-	-	-	-	-
1	0	0	0	0	0	0	0	0	0	0	0	0	0	-
1	0	1	0	0	-	-	0	0	0	0	0	0	0	-
1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	1	0	0	-	-	0	0	0	0	0	0	0	1

**Remark** X: Does not have to be set. These pins are used as external bus interface pins.  
 -: External bus interface is not used. These pins can be used as port pins.

## 5.4 Number of Instruction Wait Clocks or Data Access

Wait clocks are added to the number of clocks of an instruction when the external bus interface is accessed.

The actual number of operating clocks is therefore the sum of the number of operating clocks of each instruction and the number of wait states.

CLKOUT Pin Selection Clock	Number of Waits States (Read/Write)
f <sub>CLK</sub>	3 clocks
f <sub>CLK</sub> /2	5 or 6 clocks
f <sub>CLK</sub> /3	7 to 9 clocks
f <sub>CLK</sub> /4	9 to 12 clocks

**Remark** 1 clock: 1/f<sub>CLK</sub> (f<sub>CLK</sub>: CPU clock)

## 5.5 Number of Instruction Execution Clocks and Instruction Wait Clocks for Fetch Access

The internal flash captures an opcode every 32 bits. However, an opcode is captured from the external bus interface every 16 bits. Consequently, it takes time about two times longer than the internal flash to capture an instruction. Furthermore, a wait is inserted when accessing an external memory. Consequently, the minimum and maximum numbers of execution clocks of each instruction when fetching instructions from the external memory are as follows, for the number of clocks when instructions are fetched from the internal ROM (flash memory) area.

No. of Instruction Execution Clocks When Fetching Instructions from Internal ROM Area <sup>Note</sup>	When Fetching Instructions from External Memory	
	Minimum No. of Execution Clocks	Maximum No. of Execution Clocks
1	2 + 2 × Wait	5 + 3 × Wait
2	6 + 2 × Wait	7 + 6 × Wait
3	4 + 2 × Wait	8 + 8 × Wait
4	8 + 2 × Wait	10 + 10 × Wait
5	6 + 2 × Wait	12 + 9 × Wait
6	10 + 5 × Wait	14 + 11 × Wait

**Note** Number of clocks when the internal RAM area, SFR area, or expanded SFR area has been accessed, or when an instruction that does not access data is executed

Furthermore, the number of waits is as follows, depending on the clock selected for the CLKOUT pin.

CLKOUT Pin Selection Clock	Number of Wait States (Fetch)
f <sub>CLK</sub>	3 clocks
f <sub>CLK</sub> /2	5 or 6 clocks
f <sub>CLK</sub> /3	7 to 9 clocks
f <sub>CLK</sub> /4	9 to 12 clocks

**Caution** The flash memory and external memory are located in consecutive spaces, but start fetching in the external memory space by using a branch instruction (CALL, BR) in the flash memory or RAM memory.

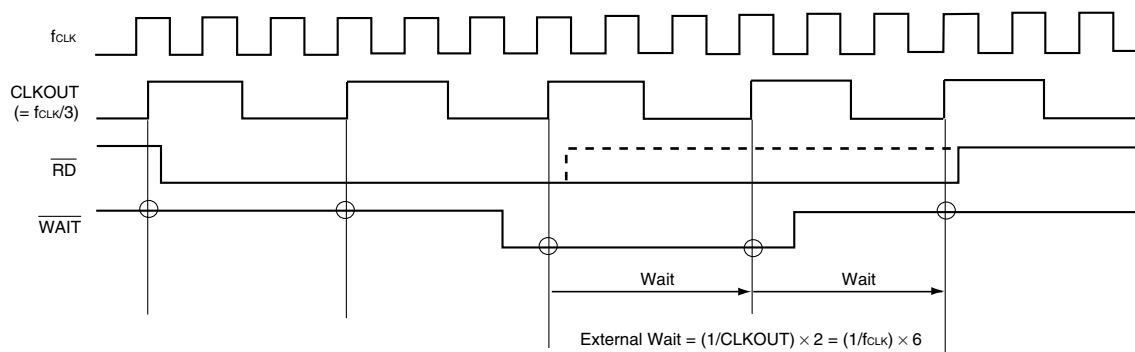
**Remark** 1 clock: 1/f<sub>CLK</sub> (f<sub>CLK</sub>: CPU clock)

## <R> 5.6 Number of Instructed Wait Cycles According to External Wait Cycles

If an external device that has a low access speed is accessed, wait cycles can be inserted into the bus cycle. If a low-level signal<sup>Note</sup> is input to the  $\overline{\text{WAIT}}$  pin at the rising edge of the CLKOUT signal, a wait of one CLKOUT clock cycle is inserted.

For example, if  $f_{\text{CLK}}/3$  is selected as the clock signal for the CLKOUT pin, and a low-level signal is input to the  $\overline{\text{WAIT}}$  pin at the rising edge of the CLKOUT signal for the second time in a row, a wait cycle of two CLKOUT clock cycles occurs. Therefore, the number of external wait cycles in  $1/f_{\text{CLK}}$  units is six clock cycles (see **Figure 5-4**).

**Figure 5-4. Example in Which External Wait Cycles Are Inserted When Separate Bus Is Read**



**Note** The  $\overline{\text{WAIT}}$  setup time from  $\text{CLKOUT} \uparrow$  ( $t_{\text{SKWT1}}$ ) and  $\overline{\text{WAIT}}$  hold time from  $\text{CLKOUT} \uparrow$  ( $t_{\text{HKWT1}}$ ) in the electrical specifications must be satisfied (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)** and **CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)**).

- Remarks**
1. 1 clock:  $1/f_{\text{CLK}}$  ( $f_{\text{CLK}}$ : CPU clock)
  2. The broken line in Figure 5-4 shows the waveform of the signal output from the  $\overline{\text{RD}}$  pin if no external wait cycle occurred.



## 5.7 Timing of External Bus Interface Function

The functions of the timing control signal output pins in the external memory extension mode are described below.

**(1)  $\overline{RD}$  pin (alternate function: P64)**

This pin outputs a read strobe signal when an instruction is fetched or data is read from the external memory. It does not output the read strobe signal (holds the high level) when the internal memory is read.

**(2)  $\overline{WR0}$  pin (alternate function: P65)**

This pin outputs a write strobe signal (in 8-bit bus mode or 16-bit bus mode (lower byte)) when data is written to the external memory.

It does not output the write strobe signal (holds the high level) when data is written to the internal memory.

**(3)  $\overline{WR1}$  pin (alternate function: P66)**

This pin outputs a write strobe signal (in 16-bit bus mode (higher byte)) when data is written to the external memory.

It does not output the write strobe signal (holds the high level) when data is written to the internal memory.

**(4)  $\overline{WAIT}$  pin (alternate function: P06)**

This pin inputs an external wait signal.

A wait can be inserted to the bus cycle by inputting an external wait signal in synchronization with the CLKOUT signal.

It can be used as an I/O port pin when the external wait signal is not used.

The external wait signal is ignored when the internal memory is accessed.

**(5)  $\overline{ASTB}$  pin (alternate function: P67)**

This pin outputs an address strobe signal.

This pin outputs the address strobe signal in the multiplexed bus mode.

During internal memory access, the address strobe signal is not output (the low level is maintained).

**(6) CLKOUT pin (alternate function: P05)**

This pin outputs the internal system clock. The internal system clock is output when the external bus interface is used (EXEN bit of the MEM register = 1).

**(7) EX0 to EX7, EX8 to EX15, EX16 to EX23, EX24 to EX31, and EX32 to EX35 pins (alternate function: P80 to P87, P50 to P57, P70 to P77, P10 to P17, and P90 to P93)**

These pins output an address signal and input/output a data signal. A valid signal is output or input when an instruction is fetched from or data is accessed to/from the external memory.

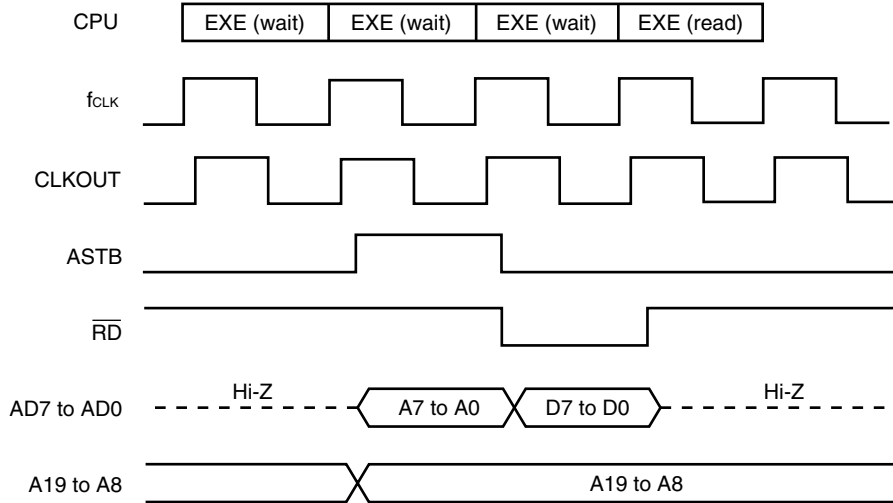
During internal memory access, the address output pin holds the address that was accessed last. The data output pin goes into the Hi-Z state.

Figures 5-5 to 5-8 show the timing charts.

5.7.1 Multiplexed bus mode

Figure 5-5. Timing to Read External Memory (1/2)

(a) No wait, 8-bit bus CLKOUT = f<sub>CLK</sub> (EXWEN = 0, MM3 = 0, MM2 = 0)



(b) With wait, 8-bit bus CLKOUT = f<sub>CLK</sub> (EXWEN = 1, MM3 = 0, MM2 = 0)

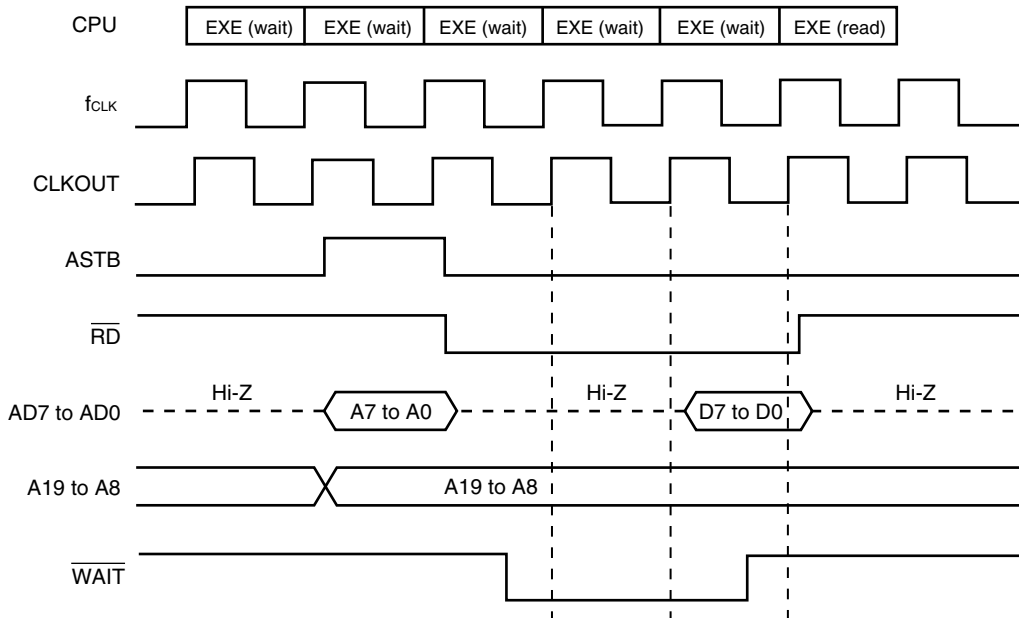
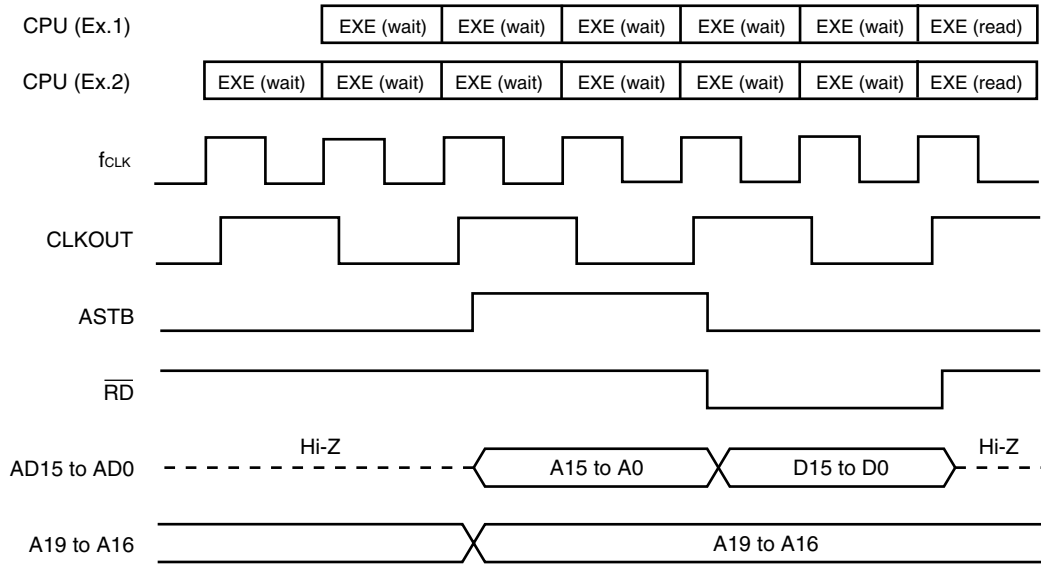


Figure 5-5. Timing to Read External Memory (2/2)

(c) No wait, 16-bit bus CLKOUT =  $f_{CLK}/2$  (EXWEN = 0, MM3 = 0, MM2 = 1)



(d) With wait, 16-bit bus CLKOUT =  $f_{CLK}/2$  (EXWEN = 1, MM3 = 0, MM2 = 1)

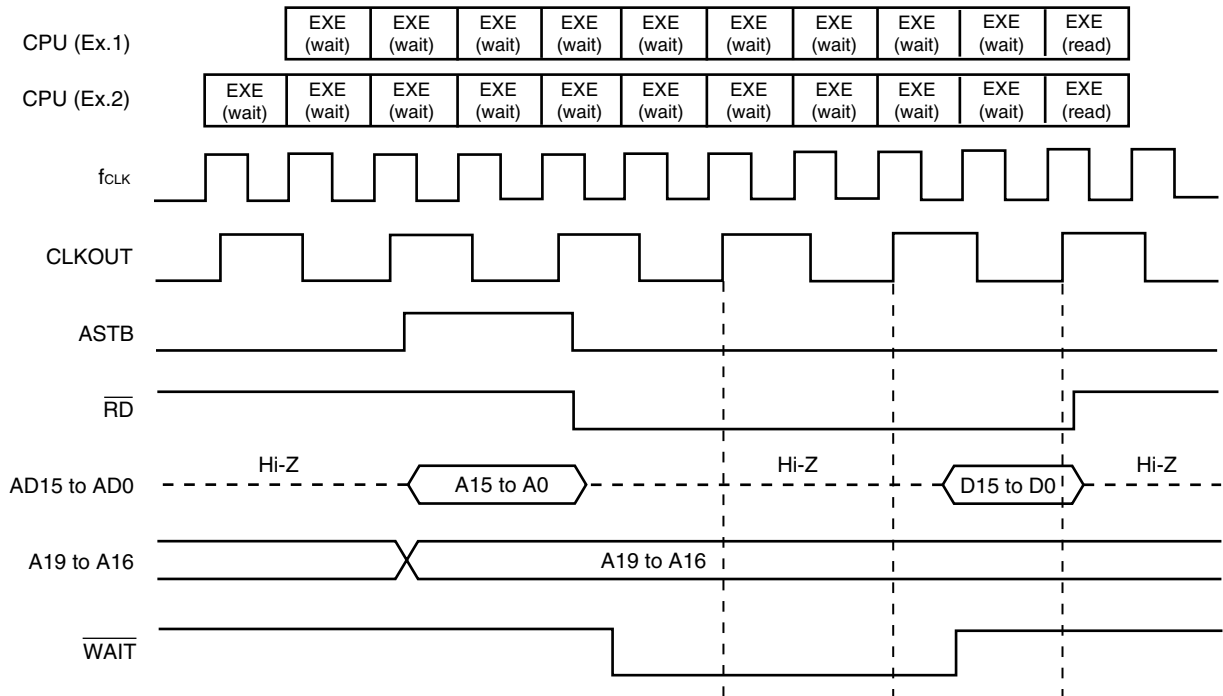
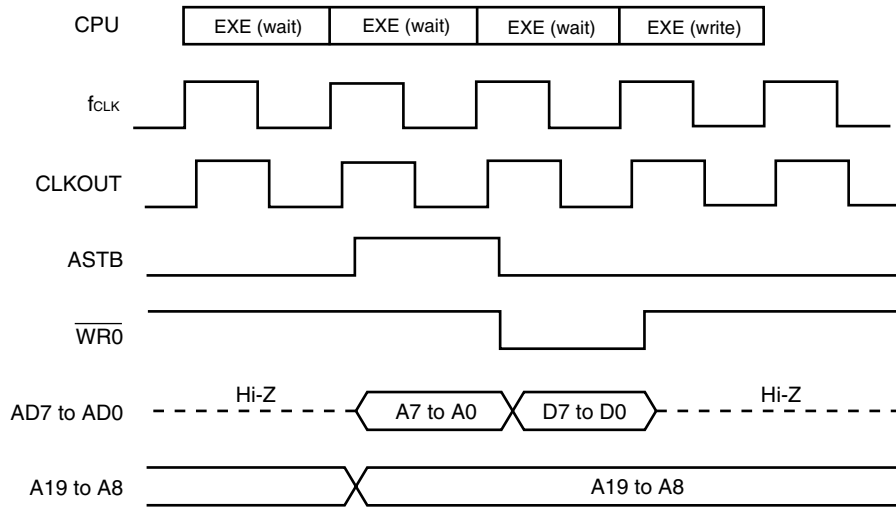


Figure 5-6. Timing to Write to External Memory (1/2)

(a) No wait, 8-bit bus CLKOUT = f<sub>CLK</sub> (EXWEN = 0, MM3 = 0, MM2 = 0)



(b) With wait, 8-bit bus CLKOUT = f<sub>CLK</sub> (EXWEN = 1, MM3 = 0, MM2 = 0)

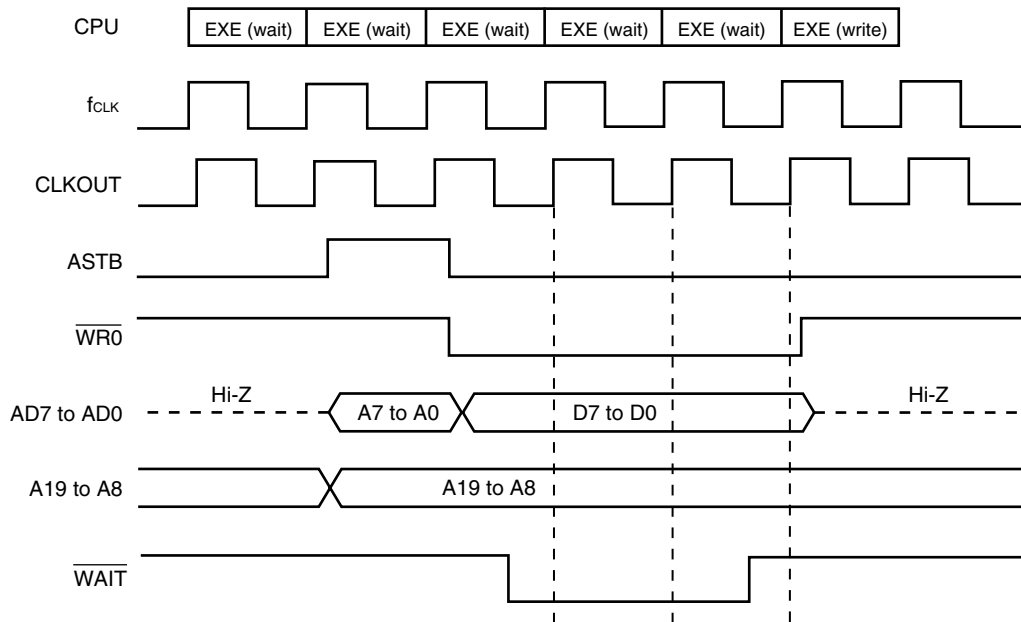
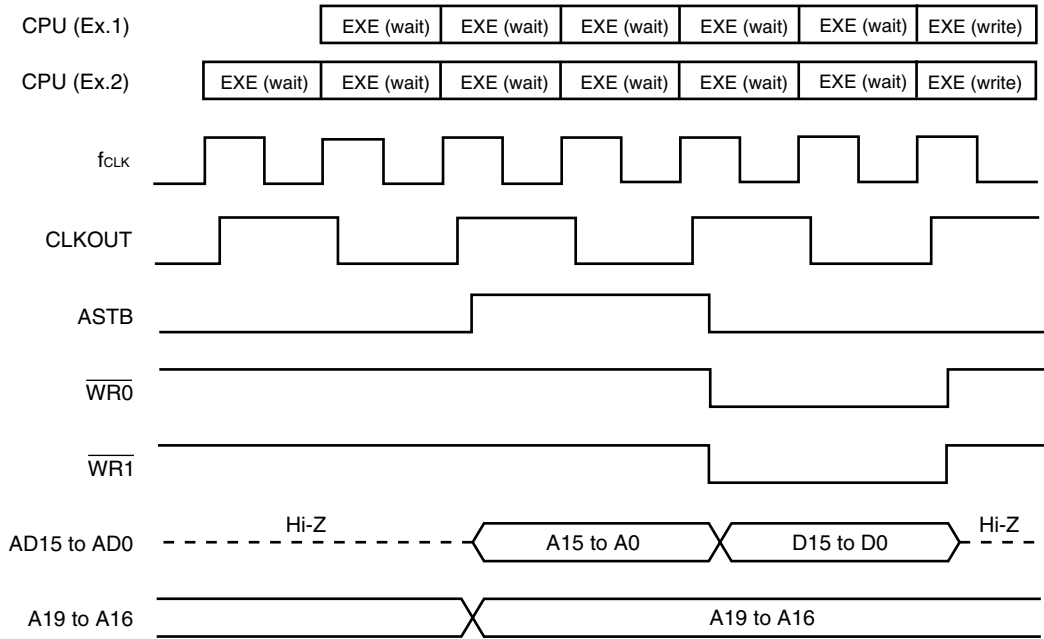
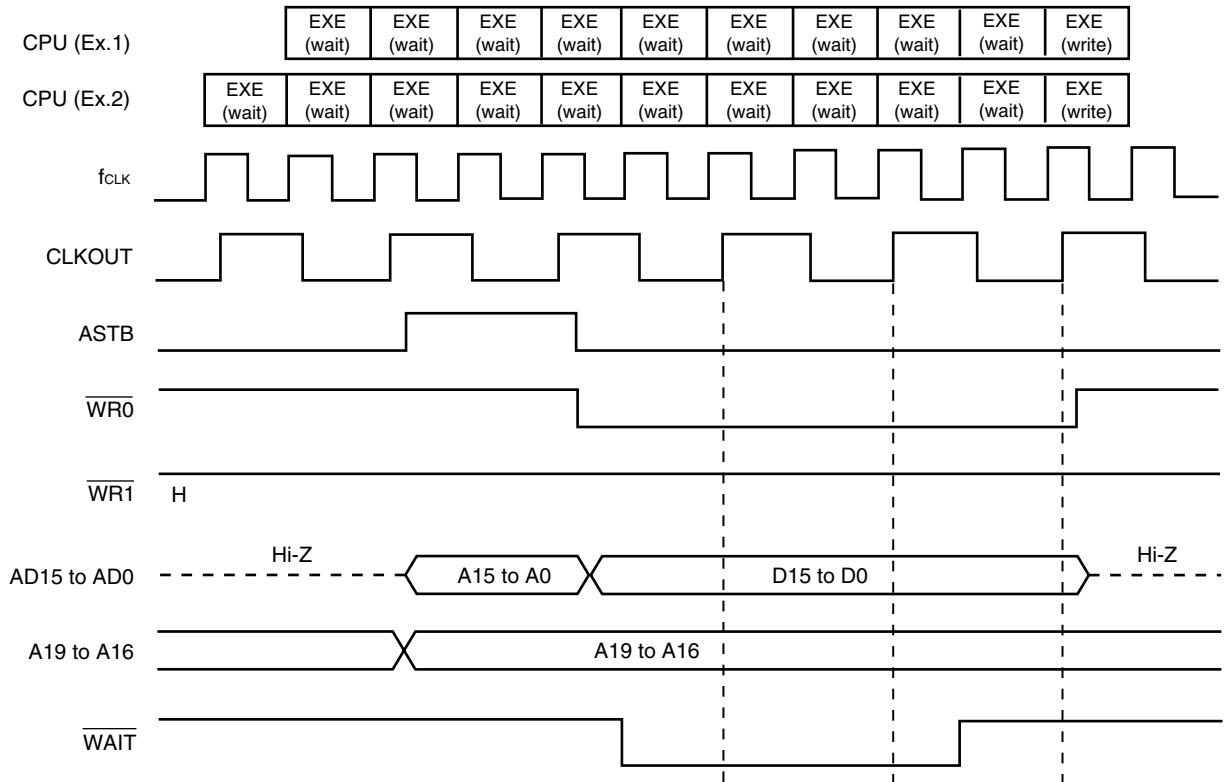


Figure 5-6. Timing to Write to External Memory (2/2)

(c) No wait, 16-bit bus CLKOUT =  $f_{CLK}/2$  (EXWEN = 0, MM3 = 0, MM2 = 1)



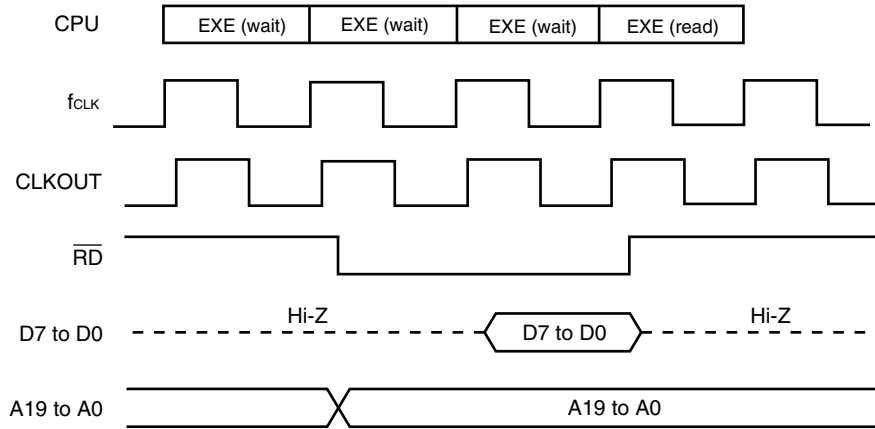
(d) With wait, 16-bit bus CLKOUT =  $f_{CLK}/2$  (EXWEN = 1, MM3 = 0, MM2 = 1), lower 8-bit writing



5.7.2 Separate bus mode

Figure 5-7. Timing to Read External Memory (1/2)

(a) No waits, 8-bit bus CLKOUT = f<sub>CLK</sub> (EXWEN = 0, MM3 = 1, MM2 = 0)



(b) With wait, 8-bit bus CLKOUT = f<sub>CLK</sub> (EXWEN = 1, MM3 = 1, MM2 = 0)

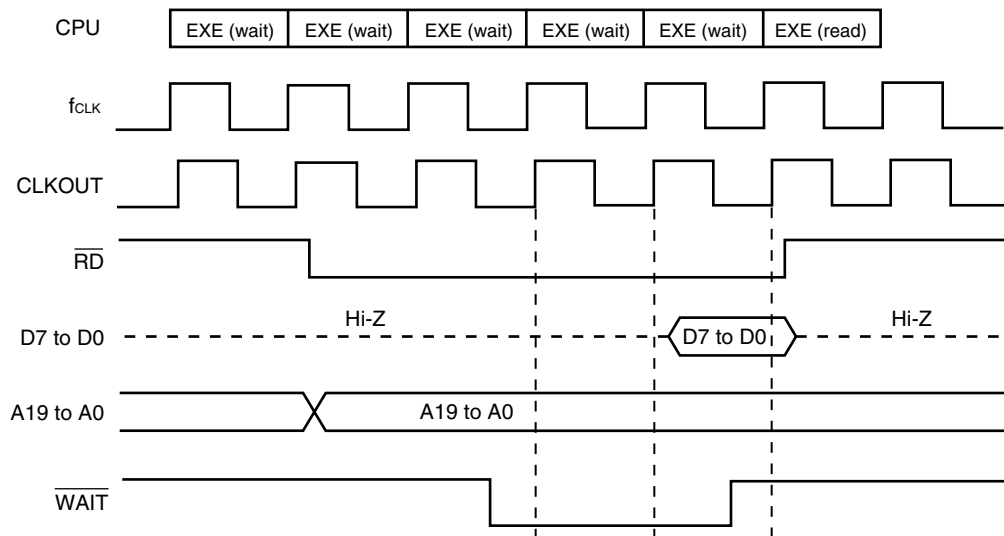
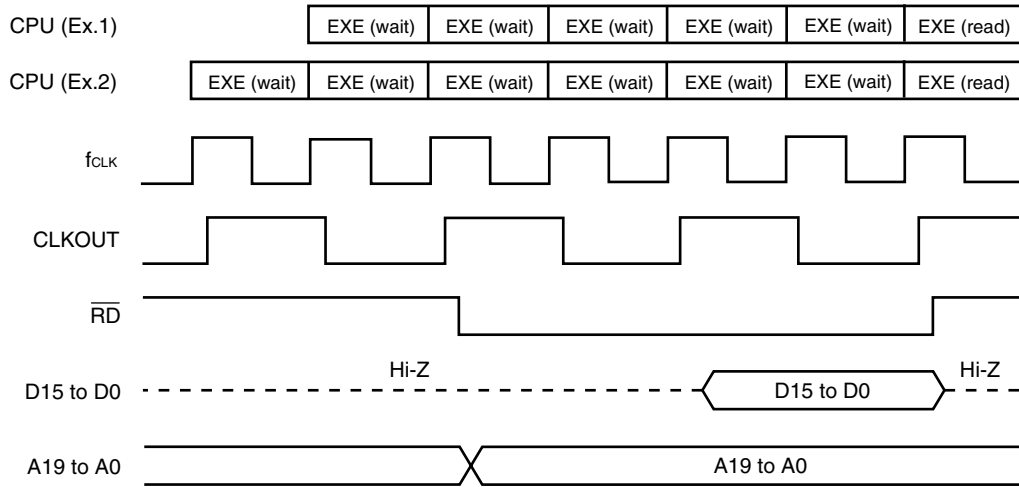


Figure 5-7. Timing to Read External Memory (2/2)

(c) No wait, 16-bit bus CLKOUT =  $f_{CLK}/2$  (EXWEN = 0, MM3 = 1, MM2 = 1)



(d) With wait, 16-bit bus CLKOUT =  $f_{CLK}/2$  (EXWEN = 1, MM3 = 1, MM2 = 1)

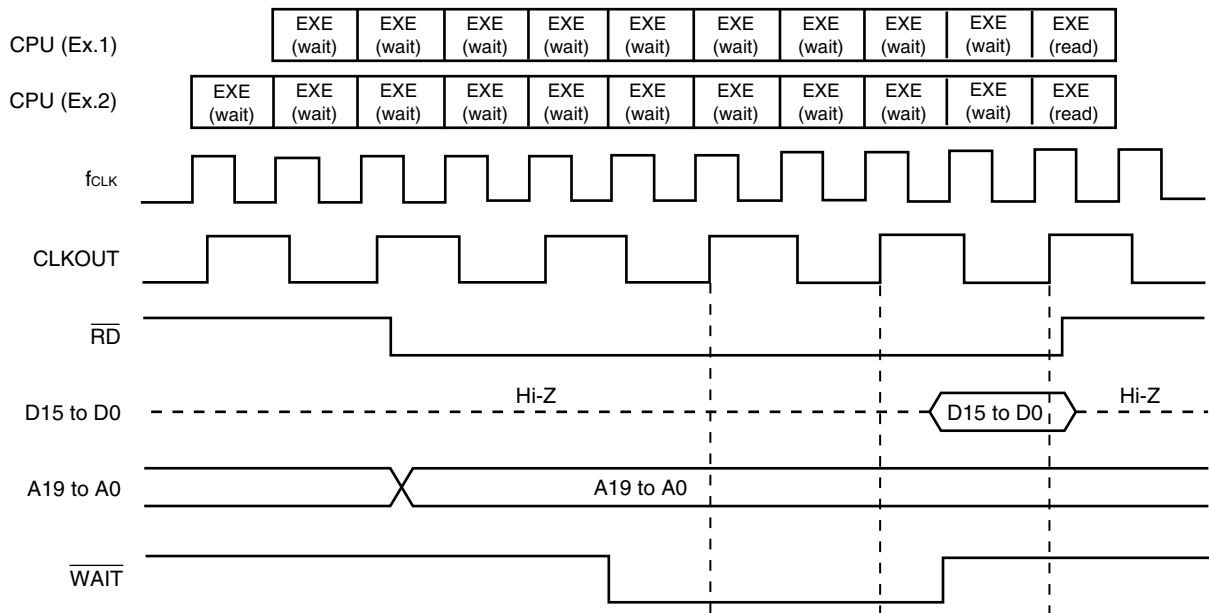
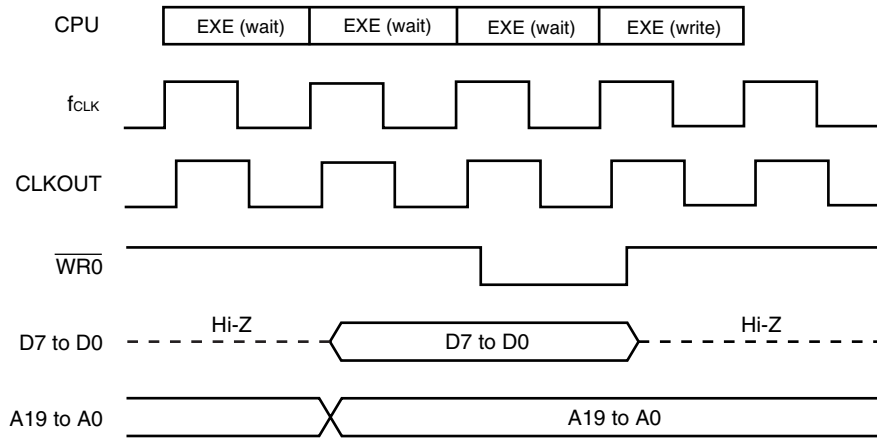


Figure 5-8. Timing to Write to External Memory (1/2)

(a) No wait, 8-bit bus CLKOUT = f<sub>CLK</sub> (EXWEN = 0, MM3 = 1, MM2 = 0)



(b) With wait, 8-bit bus CLKOUT = f<sub>CLK</sub> (EXWEN = 1, MM3 = 1, MM2 = 0)

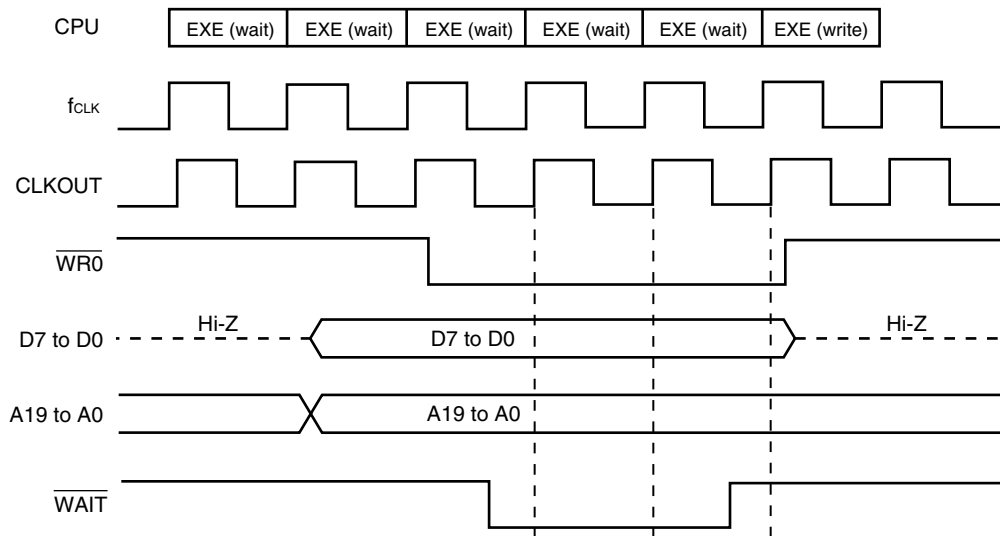
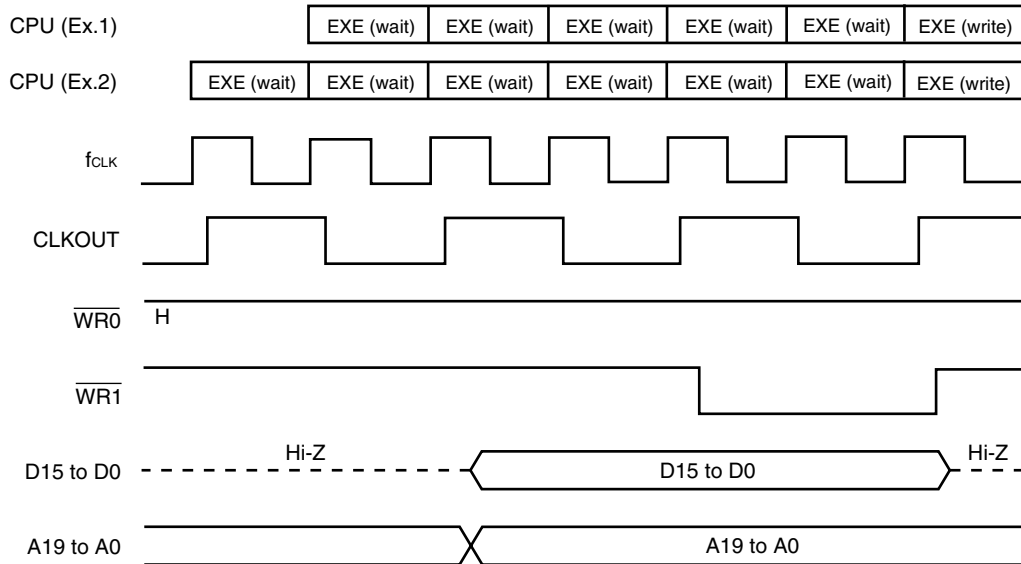


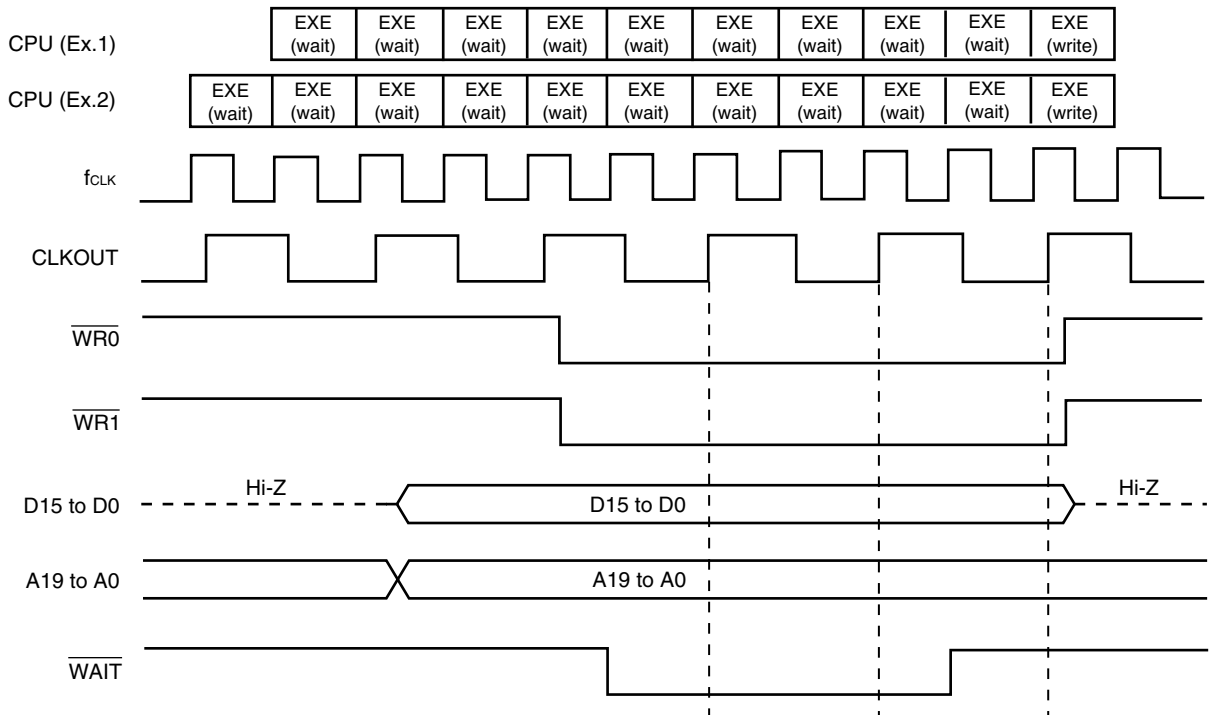


Figure 5-8. Timing to Write to External Memory (2/2)

(c) No wait, 16-bit bus CLKOUT =  $f_{CLK}/2$  (EXWEN = 0, MM3 = 1, MM2 = 1), higher 8-bit writing



(d) With wait, 16-bit bus CLKOUT =  $f_{CLK}/2$  (EXWEN = 1, MM3 = 1, MM2 = 1)

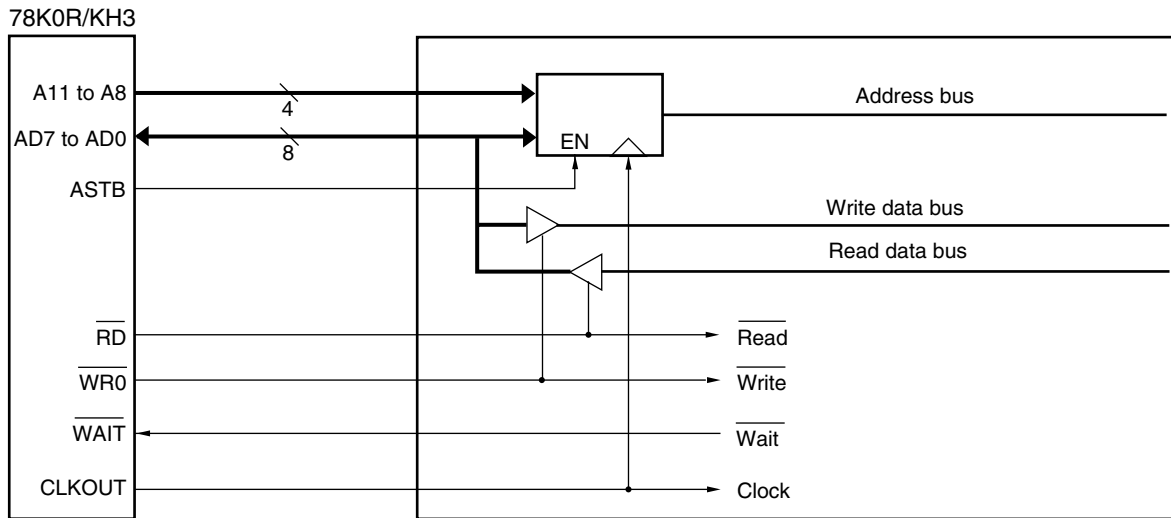


## 5.8 Example of Connection to Memory

### 5.8.1 Connection of external logic (ASIC, etc.)

When connecting the external logic, select the multiplexed bus mode or separate bus mode. When connecting buses, use CLKOUT as the reference clock. Other signals have delay on CLKOUT, however, note with caution that the CLKOUT does not delay on-board or when designing external logic.

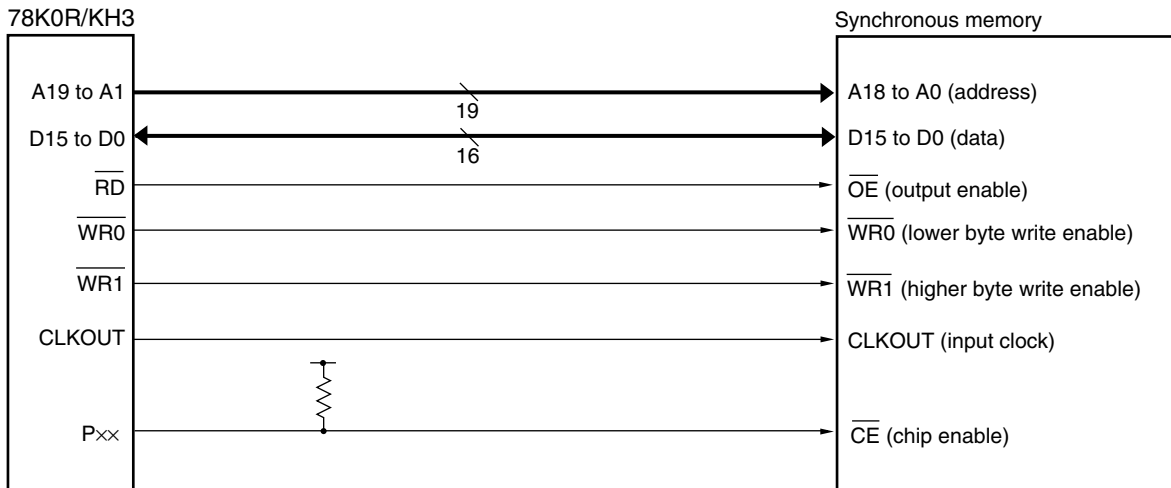
Figure 5-9. Example of External Logic Connection



### 5.8.2 Connection of synchronous memory

Use a separate bus mode for connecting a synchronous memory.

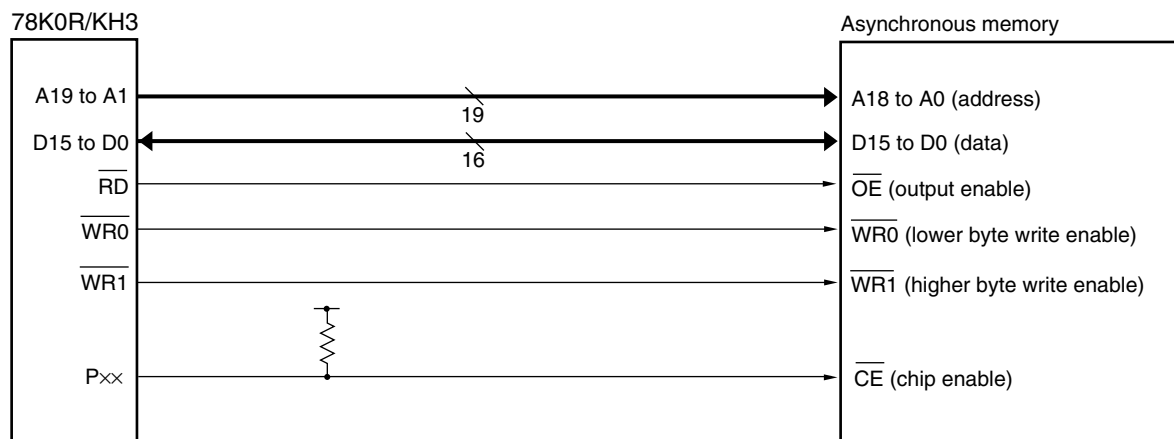
Figure 5-10. Example of Synchronous Memory Connection



### 5.8.3 Connection of asynchronous memory

Use the separate bus mode for connecting an asynchronous memory.

Figure 5-11. Example of Asynchronous Memory Connection



## CHAPTER 6 CLOCK GENERATOR

### 6.1 Functions of Clock Generator

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

#### (1) Main system clock

##### <1> X1 oscillator

This circuit oscillates a clock of  $f_x = 2$  to 20 MHz by connecting a resonator to X1 and X2.

Oscillation can be stopped by executing the STOP instruction or setting of MSTOP (bit 7 of the clock operation status control register (CSC)).

##### <2> Internal high-speed oscillator

This circuit oscillates a clock of  $f_{IH} = 8$  MHz (TYP.). After a reset release, the CPU always starts operating with this internal high-speed oscillation clock. Oscillation can be stopped by executing the STOP instruction or setting of HIOSTOP (bit 0 of CSC).

An external main system clock ( $f_{EX} = 2$  to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of MSTOP.

As the main system clock, a high-speed system clock (X1 clock or external main system clock) or internal high-speed oscillation clock can be selected by setting of MCM0 (bit 4 of the system clock control register (CKC)).

#### (2) Subsystem clock

##### • XT1 clock oscillator

This circuit oscillates a clock of  $f_{SUB} = 32.768$  kHz by connecting a 32.768 kHz resonator to XT1 and XT2. Oscillation can be stopped by setting XTSTOP (bit 6 of CSC).

#### (3) Internal low-speed oscillation clock (clock for watchdog timer)

##### • Internal low-speed oscillator

This circuit oscillates a clock of  $f_{IL} = 240$  kHz (TYP.).

The internal low-speed oscillation clock cannot be used as the CPU clock. The only hardware that operates with the internal low-speed oscillation clock is the watchdog timer.

Oscillation is stopped when the watchdog timer stops.

**Remarks 1.**  $f_x$ : X1 clock oscillation frequency  
 $f_{IH}$ : Internal high-speed oscillation clock frequency  
 $f_{EX}$ : External main system clock frequency  
 $f_{SUB}$ : Subsystem clock frequency  
 $f_{IL}$ : Internal low-speed oscillation clock frequency

**2.** The watchdog timer stops in the following cases.

- When bit 4 (WDTON) of an option byte (000C0H) = 0
- If the HALT or STOP instruction is executed when bit 4 (WDTON) of an option byte (000C0H) = 1 and bit 0 (WDSTBYON) = 0

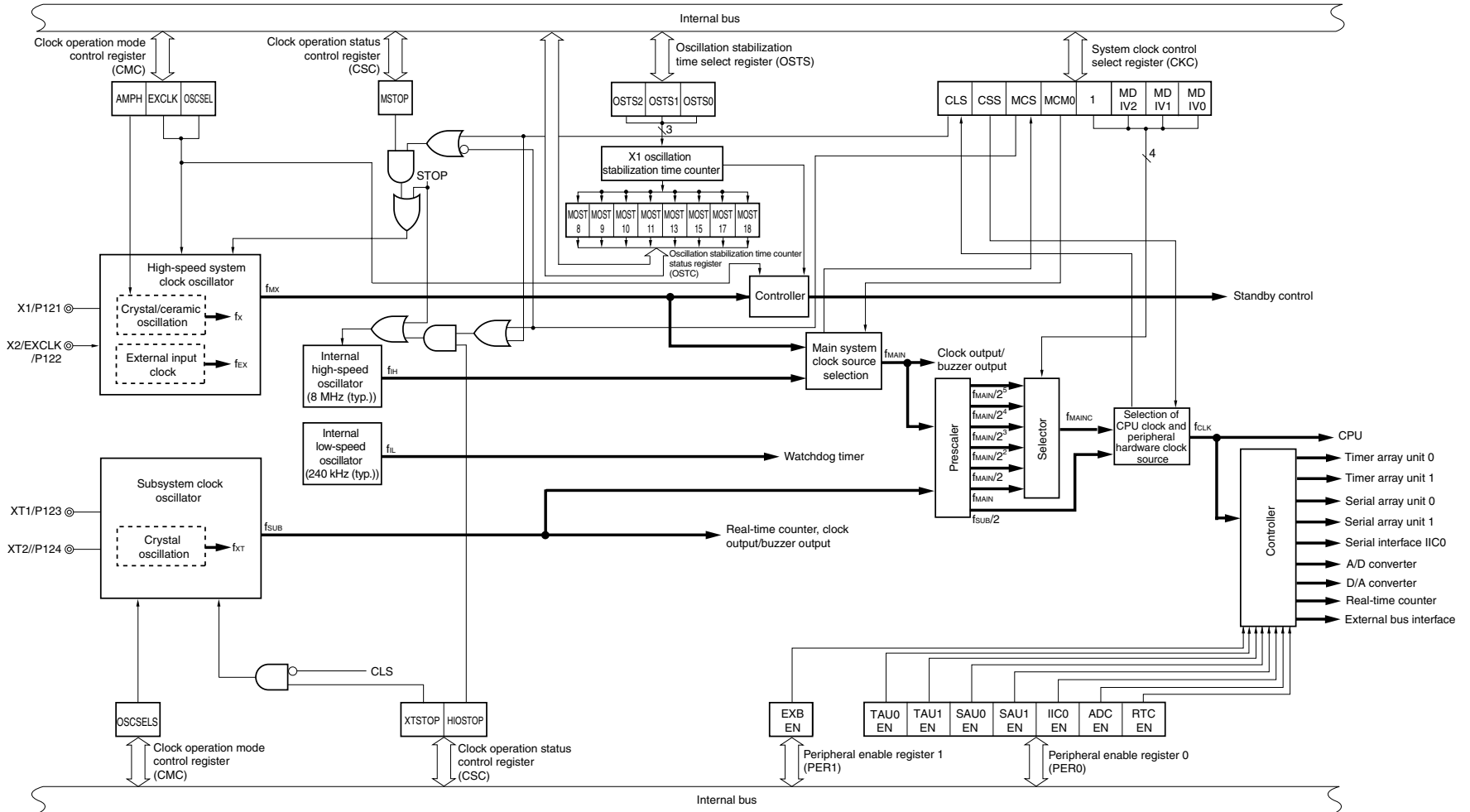
## 6.2 Configuration of Clock Generator

The clock generator includes the following hardware.

**Table 6-1. Configuration of Clock Generator**

Item	Configuration
Control registers	Clock operation mode control register (CMC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) System clock control register (CKC) Peripheral enable registers 0, 1 (PER0, PER1) Operation speed mode control register (OSMC) Internal high-speed oscillator trimming register (HIOTRM)
Oscillators	X1 oscillator XT1 oscillator Internal high-speed oscillator Internal low-speed oscillator

<R> Figure 6-1. Block Diagram of Clock Generator



&lt;R&gt;

<b>Remark</b>	f <sub>X</sub> :	X1 clock oscillation frequency
	f <sub>IH</sub> :	Internal high-speed oscillation clock frequency
	f <sub>EX</sub> :	External main system clock frequency
	f <sub>MX</sub> :	High-speed system clock frequency
	f <sub>MAIN</sub> :	Main system clock frequency
	f <sub>MAINC</sub> :	Main system select clock frequency
	f <sub>XT</sub> :	XT1 clock oscillation frequency
	f <sub>SUB</sub> :	Subsystem clock frequency
	f <sub>CLK</sub> :	CPU/peripheral hardware clock frequency
	f <sub>IL</sub> :	Internal low-speed oscillation clock frequency

### 6.3 Registers Controlling Clock Generator

The following eight registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- System clock control register (CKC)
- Peripheral enable registers 0, 1 (PER0, PER1)
- Operation speed mode control register (OSMC)
- Internal high-speed oscillator trimming register (HIOTRM)

**(1) Clock operation mode control register (CMC)**

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

CMC can be written only once by an 8-bit memory manipulation instruction after reset release. This register can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 6-2. Format of Clock Operation Mode Control Register (CMC)**

Address: FFFA0H After reset: 00H R/W

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

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

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

AMPH	Control of X1 clock oscillation frequency
0	$2\text{ MHz} \leq f_x \leq 10\text{ MHz}$
1	$10\text{ MHz} < f_x \leq 20\text{ MHz}$

<R>

- Cautions**
1. CMC can be written only once after reset release, by an 8-bit memory manipulation instruction.
  2. After reset release, set CMC before X1 or XT1 oscillation is started as set by the clock operation status control register (CSC).
  3. Be sure to set AMPH to 1 if the X1 clock oscillation frequency exceeds 10 MHz.
  4. It is recommended to set the default value (00H) to CMC after reset release, even when the register is used at the default value, in order to prevent malfunctioning during a program loop.

**Remark** fx: X1 clock oscillation frequency



**(2) Clock operation status control register (CSC)**

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

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

Reset signal generation sets this register to C0H.

**Figure 6-3. Format of Clock Operation Status Control Register (CSC)**

Address: FFFA1H After reset: C0H R/W

Symbol	<7>	<6>	5	4	3	2	1	<0>
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
MSTOP	Control of high-speed system clock operation							
	X1 oscillation mode		External clock input mode		Input port mode			
0	X1 oscillator operating		External clock from EXCLK pin is valid		-			
1	X1 oscillator stopped		External clock from EXCLK pin is invalid					
XTSTOP	Subsystem clock operation control							
	XT1 oscillation mode			Input port mode				
0	XT1 oscillator operating			-				
1	XT1 oscillator stopped							
HIOSTOP								
0	Internal high-speed oscillator operating							
1	Internal high-speed oscillator stopped							

- Cautions**
1. After reset release, set the clock operation mode control register (CMC) before starting X1 oscillation as set by MSTOP or XT1 oscillation as set by XTSTOP.
  2. To start X1 oscillation as set by MSTOP, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).
  3. Do not stop the clock selected for the CPU/peripheral hardware clock ( $f_{CLK}$ ) with the CSC register.
  4. The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) and the condition before clock oscillation is to be stopped are as shown in Table 6-2.

**Table 6-2. Condition Before Stopping Clock Oscillation and Flag Setting**

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock	<ul style="list-style-type: none"> <li>• CLS = 0 and MCS = 0</li> <li>• CLS = 1</li> </ul> (CPU and peripheral hardware clocks operate with a clock other than the high-speed system clock.)	MSTOP = 1
External main system clock		
Subsystem clock	<ul style="list-style-type: none"> <li>• CLS = 0</li> </ul> (CPU and peripheral hardware clocks operate with a clock other than the subsystem clock.)	XTSTOP = 1
Internal high-speed oscillation clock	<ul style="list-style-type: none"> <li>• CLS = 0 and MCS = 1</li> <li>• CLS = 1</li> </ul> (CPU and peripheral hardware clocks operate with a clock other than the internal high-speed oscillator clock.)	HIOSTOP = 1

**(3) Oscillation stabilization time counter status register (OSTC)**

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

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

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

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

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

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

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

Figure 6-4. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H After reset: 00H R

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

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

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

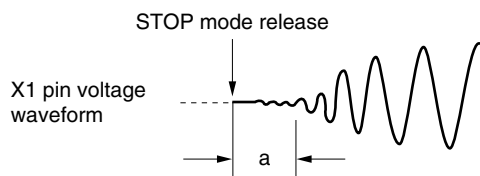
**2.** The oscillation stabilization time counter counts up to the oscillation stabilization time set by OST8.

In the following cases, set the oscillation stabilization time of OST8 to the value greater than or equal to the count value which is to be checked by the OSTC register.

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

(Note, therefore, that only the status up to the oscillation stabilization time set by OST8 is set to OSTC after the STOP mode is released.)

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



**Remark** fx: X1 clock oscillation frequency

**(4) Oscillation stabilization time select register (OSTS)**

This register is used to select the X1 clock oscillation stabilization wait time when the STOP mode is released.

When the X1 clock is selected as the CPU clock, the operation automatically waits for the time set using OSTS after the STOP mode is released.

When the internal high-speed oscillation clock is selected as the CPU clock, confirm with OSTC that the desired oscillation stabilization time has elapsed after the STOP mode is released. The oscillation stabilization time can be checked up to the time set using OSTC.

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

Reset signal generation sets OSTS to 07H.

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

Address: FFFA3H After reset: 07H R/W

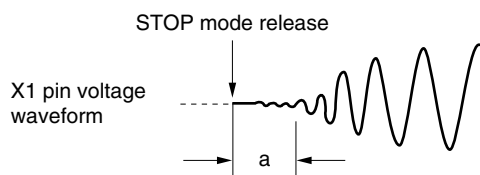
Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection	Oscillation stabilization time selection	
				$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$
0	0	0	$2^9/f_x$	25.6 $\mu\text{s}$	Setting prohibited
0	0	1	$2^9/f_x$	51.2 $\mu\text{s}$	25.6 $\mu\text{s}$
0	1	0	$2^{10}/f_x$	102.4 $\mu\text{s}$	51.2 $\mu\text{s}$
0	1	1	$2^{11}/f_x$	204.8 $\mu\text{s}$	102.4 $\mu\text{s}$
1	0	0	$2^{13}/f_x$	819.2 $\mu\text{s}$	409.6 $\mu\text{s}$
1	0	1	$2^{15}/f_x$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_x$	13.11 ms	6.55 ms
1	1	1	$2^{19}/f_x$	26.21 ms	13.11 ms

- Cautions**
- To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTS register before executing the STOP instruction.
  - Setting the oscillation stabilization time to 20  $\mu\text{s}$  or less is prohibited.
  - To change the setting of the OSTS register, be sure to confirm that the counting operation of the OSTC register has been completed.
  - Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.
  - The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS.

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

- If the X1 clock starts oscillation while the internal high-speed oscillation clock or subsystem clock is being used as the CPU clock.
  - If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock with the X1 clock oscillating. (Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after the STOP mode is released.)
- The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



**Remark**  $f_x$ : X1 clock oscillation frequency

**(5) System clock control register (CKC)**

This register is used to select a CPU/peripheral hardware clock and a division ratio.  
 CKC can be set by a 1-bit or 8-bit memory manipulation instruction.  
 Reset signal generation sets this register to 09H.

**Figure 6-6. Format of System Clock Control Register (CKC)**

Address: FFFA4H After reset: 09H R/W<sup>Note 1</sup>

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	1	MDIV2	MDIV1	MDIV0

CLS	Status of CPU/peripheral hardware clock (f <sub>CLK</sub> )
0	Main system clock (f <sub>MAIN</sub> )
1	Subsystem clock (f <sub>SUB</sub> )

MCS	Status of Main system clock (f <sub>MAIN</sub> )
0	Internal high-speed oscillation clock (f <sub>IH</sub> )
1	High-speed system clock (f <sub>MX</sub> )

CSS	MCM0	MDIV2	MDIV1	MDIV0	Selection of CPU/peripheral hardware clock (f <sub>CLK</sub> )
0	0	0	0	0	f <sub>IH</sub>
		0	0	1	f <sub>IH</sub> /2 (default)
		0	1	0	f <sub>IH</sub> /2 <sup>2</sup>
		0	1	1	f <sub>IH</sub> /2 <sup>3</sup>
		1	0	0	f <sub>IH</sub> /2 <sup>4</sup>
		1	0	1	f <sub>IH</sub> /2 <sup>5</sup>
0	1	0	0	0	f <sub>MX</sub>
		0	0	1	f <sub>MX</sub> /2
		0	1	0	f <sub>MX</sub> /2 <sup>2</sup>
		0	1	1	f <sub>MX</sub> /2 <sup>3</sup>
		1	0	0	f <sub>MX</sub> /2 <sup>4</sup>
		1	0	1	f <sub>MX</sub> /2 <sup>5</sup> <sup>Note 2</sup>
1 <sup>Note 3</sup>	× <sup>Note 3</sup>	×	×	×	f <sub>SUB</sub> /2
Other than above					Setting prohibited

- Notes**
1. Bits 7 and 5 are read-only.
  2. Setting is prohibited when f<sub>MX</sub> < 4 MHz.
  3. Changing the value of the MCM0 bit is prohibited while CSS is set to 1.

- Remarks**
1. f<sub>IH</sub>: Internal high-speed oscillation clock frequency  
 f<sub>MX</sub>: High-speed system clock frequency  
 f<sub>SUB</sub>: Subsystem clock frequency
  2. ×: don't care

(Cautions 1 to 3 are listed on the next page.)

- Cautions**
1. Be sure to set bit 3 to 1.
  2. The clock set by CSS, MCM0, and MDIV2 to MDIV0 is supplied to the CPU and peripheral hardware. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the real-time counter, clock output/buzzer output, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral operating hardware clock.
  3. If the peripheral hardware clock is used as the subsystem clock, the operations of the A/D converter and IIC0 are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS).

The fastest instruction can be executed in 1 clock of the CPU clock in the 78K0R/KH3. Therefore, the relationship between the CPU clock ( $f_{CLK}$ ) and the minimum instruction execution time is as shown in Table 6-3.

**Table 6-3. Relationship Between CPU Clock and Minimum Instruction Execution Time**

CPU Clock (Value set by the MDIV2 to MDIV0 bits)	Minimum Instruction Execution Time: $1/f_{CLK}$			
	Main System Clock (CSS = 0)			Subsystem Clock (CSS = 1)
	High-Speed System Clock (MCM0 = 1)		Internal High-Speed Oscillation Clock (MCM0 = 0)	
	At 10 MHz Operation	At 20 MHz Operation	At 8 MHz (TYP.) Operation	At 32.768 kHz Operation
$f_{MAIN}$	0.1 $\mu s$	0.05 $\mu s$	0.125 $\mu s$ (TYP.)	–
$f_{MAIN}/2$	0.2 $\mu s$	0.1 $\mu s$	0.25 $\mu s$ (TYP.) (default)	–
$f_{MAIN}/2^2$	0.4 $\mu s$	0.2 $\mu s$	0.5 $\mu s$ (TYP.)	–
$f_{MAIN}/2^3$	0.8 $\mu s$	0.4 $\mu s$	1.0 $\mu s$ (TYP.)	–
$f_{MAIN}/2^4$	1.6 $\mu s$	0.8 $\mu s$	2.0 $\mu s$ (TYP.)	–
$f_{MAIN}/2^5$	3.2 $\mu s$	1.6 $\mu s$	4.0 $\mu s$ (TYP.)	–
$f_{SUB}/2$	–	–	–	61 $\mu s$

**Remark**  $f_{MAIN}$ : Main system clock frequency ( $f_{IH}$  or  $f_{MX}$ )  
 $f_{SUB}$ : Subsystem clock frequency

**(6) Peripheral enable registers 0, 1 (PER0, PER1)**

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

PER0 and PER1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Figure 6-7. Format of Peripheral Enable Register (1/2)**

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

Address: F00F1H After reset: 00H R/W

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

RTCEN	Control of real-time counter (RTC) input clock <sup>Note</sup>
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by the real-time counter (RTC) cannot be written.</li> <li>• Operation of the real-time counter (RTC) is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the real-time counter (RTC) can be read and written.</li> </ul>

DACEN	Control of D/A converter input clock
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by D/A converter cannot be written.</li> <li>• The D/A converter is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the D/A converter can be read and written.</li> </ul>

ADCEN	Control of A/D converter input clock
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by the A/D converter cannot be written.</li> <li>• The A/D converter is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the A/D converter can be read and written.</li> </ul>

IIC0EN	Control of serial interface IIC0 input clock
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by the serial interface IIC0 cannot be written.</li> <li>• The serial interface IIC0 is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the serial interface IIC0 can be read and written.</li> </ul>

**Note** The input clock that can be controlled by RTCEN is used when the register that is used by the real-time counter (RTC) is accessed from the CPU. RTCEN cannot control supply of the operating clock ( $f_{SUB}$ ) to RTC.

**Caution** Be sure to clear bits 1 to 7 of PER1 to 0.



Figure 6-7. Format of Peripheral Enable Register (2/2)

SAU1EN	Control of serial array unit 1 input clock
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by the serial array unit 1 cannot be written.</li> <li>• The serial array unit 1 is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the serial array unit 1 can be read and written.</li> </ul>

SAU0EN	Control of serial array unit 0 input clock
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by the serial array unit 0 cannot be written.</li> <li>• The serial array unit 0 is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the serial array unit 0 can be read and written.</li> </ul>

TAU1EN	Control of timer array unit 1 input clock
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by the timer array unit 1 cannot be written.</li> <li>• The timer array unit 1 is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the timer array unit 1 can be read and written.</li> </ul>

TAU0EN	Control of timer array unit 0 input clock
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by the timer array unit 0 cannot be written.</li> <li>• The timer array unit 0 is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the timer array unit 0 can be read and written.</li> </ul>

EXBEN	Control of external bus interface input clock
0	Stops input clock supply. <ul style="list-style-type: none"> <li>• SFR used by the external bus interface cannot be written.</li> <li>• The external bus interface is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the external bus interface can be read and written.</li> </ul>

**Caution** Be sure to clear bits 1 to 7 of PER1 to 0.

**(7) Operation speed mode control register (OSMC)**

This register is used to control the step-up circuit of the flash memory for high-speed operation.

If the microcontroller operates at a low speed with a system clock of 10 MHz or less, the power consumption can be lowered by setting this register to the default value, 00H.

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

Reset signal generation clears this register to 00H.

**Figure 6-8. Format of Operation Speed Mode Control Register (OSMC)**

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	0	0	0	0	0	0	0	FSEL

FSEL	$f_{CLK}$ frequency selection
0	Operates at a frequency of 10 MHz or less (default).
1	Operates at a frequency higher than 10 MHz.

**Cautions 1. OSMC can be written only once after reset release, by an 8-bit memory manipulation instruction.**

**2. Write “1” to FSEL before the following two operations.**

- Changing the clock prior to dividing  $f_{CLK}$  to a clock other than  $f_{IH}$ .
- Operating the DMA controller.

**3. The CPU waits when “1” is written to the FSEL flag.**

**Interrupt requests issued during a wait will be suspended.**

**The wait time is 16.6  $\mu$ s to 18.5  $\mu$ s when  $f_{CLK} = f_{IH}$ , and 33.3  $\mu$ s to 36.9  $\mu$ s when  $f_{CLK} = f_{IH}/2$ .**

**However, counting the oscillation stabilization time of  $f_x$  can continue even while the CPU is waiting.**

**4. To increase  $f_{CLK}$  to 10 MHz or higher, set FSEL to “1”, then change  $f_{CLK}$  after two or more clocks have elapsed. Use the external bus interface two clock cycles after setting FSEL to 1.**

**5. Flash memory can be used at a frequency of 10 MHz or lower if FSEL is 1.**

<R>

<R>

**(8) Internal high-speed oscillator trimming register (HIOTRM)**

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

With self-measurement of the internal high-speed oscillator frequency via a subsystem clock using a crystal resonator, a timer using high-accuracy external clock input (real-time counter or timer array unit), and so on, the register can adjust the accuracy.

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

Reset signal generation sets this register to 10H.

**Caution** The frequency will vary if the temperature and  $V_{DD}$  pin voltage change after accuracy adjustment. Moreover, if the HIOTRM register is set to any value other than the initial value (10H), the oscillation accuracy of the internal high-speed oscillation clock may exceed  $8\text{ MHz}\pm 5\%$ , depending on the subsequent temperature and  $V_{DD}$  voltage change, or HIOTRM register setting. When the temperature and  $V_{DD}$  voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.

Figure 6-9. Format of Internal High-Speed Oscillator Trimming Register (HIOTRM)

Address: F00F2H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
HIOTRM	0	0	0	TTRM4	TTRM3	TTRM2	TTRM1	TTRM0

TTRM4	TTRM3	TTRM2	TTRM1	TTRM0	Clock correction value (2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V)		
					MIN.	TYP.	MAX.
0	0	0	0	0	-5.54%	-4.88%	-4.02%
0	0	0	0	1	-5.28%	-4.62%	-3.76%
0	0	0	1	0	-4.99%	-4.33%	-3.47%
0	0	0	1	1	-4.69%	-4.03%	-3.17%
0	0	1	0	0	-4.39%	-3.73%	-2.87%
0	0	1	0	1	-4.09%	-3.43%	-2.57%
0	0	1	1	0	-3.79%	-3.13%	-2.27%
0	0	1	1	1	-3.49%	-2.83%	-1.97%
0	1	0	0	0	-3.19%	-2.53%	-1.67%
0	1	0	0	1	-2.88%	-2.22%	-1.36%
0	1	0	1	0	-2.23%	-1.91%	-1.31%
0	1	0	1	1	-1.92%	-1.60%	-1.28%
0	1	1	0	0	-1.60%	-1.28%	-0.96%
0	1	1	0	1	-1.28%	-0.96%	-0.64%
0	1	1	1	0	-0.96%	-0.64%	-0.32%
0	1	1	1	1	-0.64%	-0.32%	±0%
1	0	0	0	0	±0% (default)		
1	0	0	0	1	+0%	+0.32%	+0.64%
1	0	0	1	0	+0.33%	+0.65%	+0.97%
1	0	0	1	1	+0.66%	+0.98%	+1.30%
1	0	1	0	0	+0.99%	+1.31%	+1.63%
1	0	1	0	1	+1.32%	+1.64%	+1.96%
1	0	1	1	0	+1.38%	+1.98%	+2.30%
1	0	1	1	1	+1.46%	+2.32%	+2.98%
1	1	0	0	0	+1.80%	+2.66%	+3.32%
1	1	0	0	1	+2.14%	+3.00%	+3.66%
1	1	0	1	0	+2.48%	+3.34%	+4.00%
1	1	0	1	1	+2.83%	+3.69%	+4.35%
1	1	1	0	0	+3.18%	+4.04%	+4.70%
1	1	1	0	1	+3.53%	+4.39%	+5.05%
1	1	1	1	0	+3.88%	+4.74%	+5.40%
1	1	1	1	1	+4.24%	+5.10%	+5.76%

**Caution** The internal high-speed oscillation frequency becomes faster/slower by increasing/decreasing the HIOTRM value to a value larger/smaller than a certain value. A reversal, such as the frequency becoming slower/faster by increasing/decreasing the HIOTRM value does not occur.

## 6.4 System Clock Oscillator

### 6.4.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (2 to 20 MHz) connected to the X1 and X2 pins.

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

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

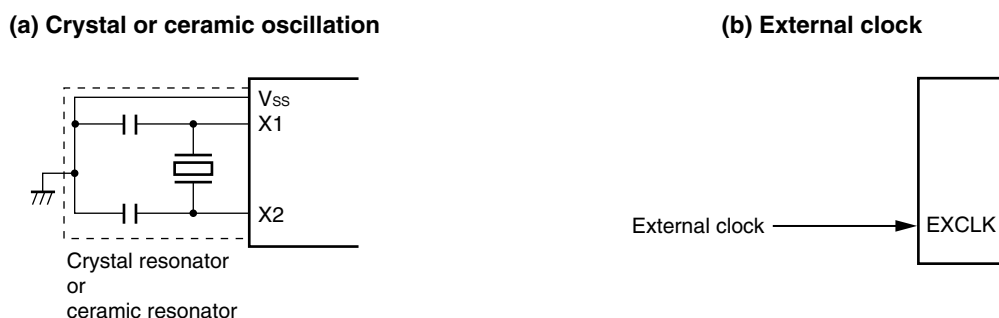
- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

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

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

Figure 6-10 shows an example of the external circuit of the X1 oscillator.

**Figure 6-10. Example of External Circuit of X1 Oscillator**



Cautions are listed on the next page.

### 6.4.2 XT1 oscillator

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

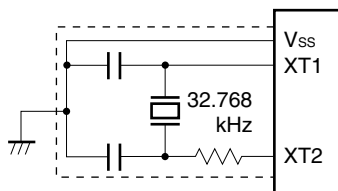
To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

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

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

Figure 6-11 shows an example of the external circuit of the XT1 oscillator.

**Figure 6-11. Example of External Circuit of XT1 Oscillator (Crystal Oscillation)**



Cautions are listed on the next page.

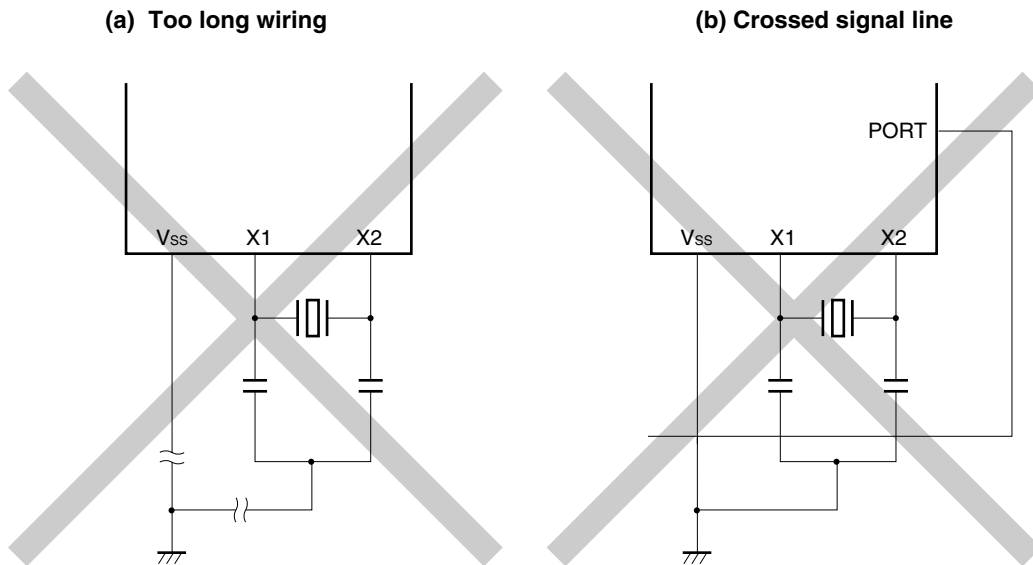
**Caution** When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 6-10 and 6-11 to avoid an adverse effect from wiring capacitance.

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

Note that the XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption.

Figure 6-12 shows examples of incorrect resonator connection.

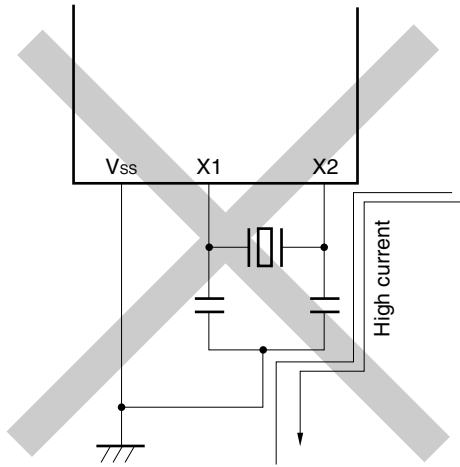
Figure 6-12. Examples of Incorrect Resonator Connection (1/2)



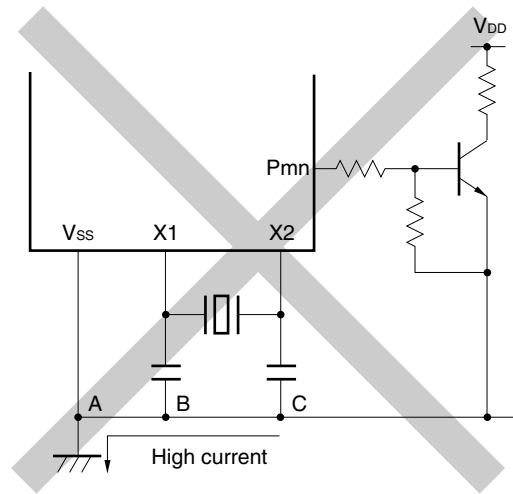
**Remark** When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 6-12. Examples of Incorrect Resonator Connection (2/2)

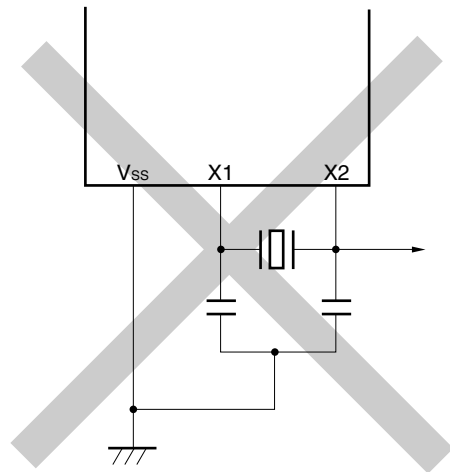
(c) Wiring near high alternating current



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



(e) Signals are fetched



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

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

### 6.4.3 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0R/KH3 (8 MHz (TYP.)). Oscillation can be controlled by bit 0 (HIOSSTOP) of the clock operation status control register (CSC).

After a reset release, the internal high-speed oscillator automatically starts oscillation.

### 6.4.4 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0R/KH3.

The internal low-speed oscillation clock is used only as the watchdog timer clock. The internal low-speed oscillation clock cannot be used as the CPU clock.

After a reset release, the internal low-speed oscillator automatically starts oscillation, and the watchdog timer is driven (240 kHz (TYP.)) if the watchdog timer operation is enabled by the option byte.

The internal low-speed oscillator continues oscillation except when the watchdog timer stops. When the watchdog timer operates, the internal low-speed oscillation clock does not stop, even in case of a program loop.

### 6.4.5 Prescaler

The prescaler generates CPU/peripheral hardware clock by dividing the main system clock and subsystem clock.



## 6.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 6-1**).

- Main system clock  $f_{MAIN}$ 
  - High-speed system clock  $f_{MX}$ 
    - X1 clock  $f_X$
    - External main system clock  $f_{EX}$
  - Internal high-speed oscillation clock  $f_{IH}$
- Subsystem clock  $f_{SUB}$
- Internal low-speed oscillation clock  $f_{IL}$
- CPU/peripheral hardware clock  $f_{CLK}$

The CPU starts operation when the internal high-speed oscillator starts outputting after a reset release in the 78K0R/KH3, thus enabling the following.

### (1) Enhancement of security function

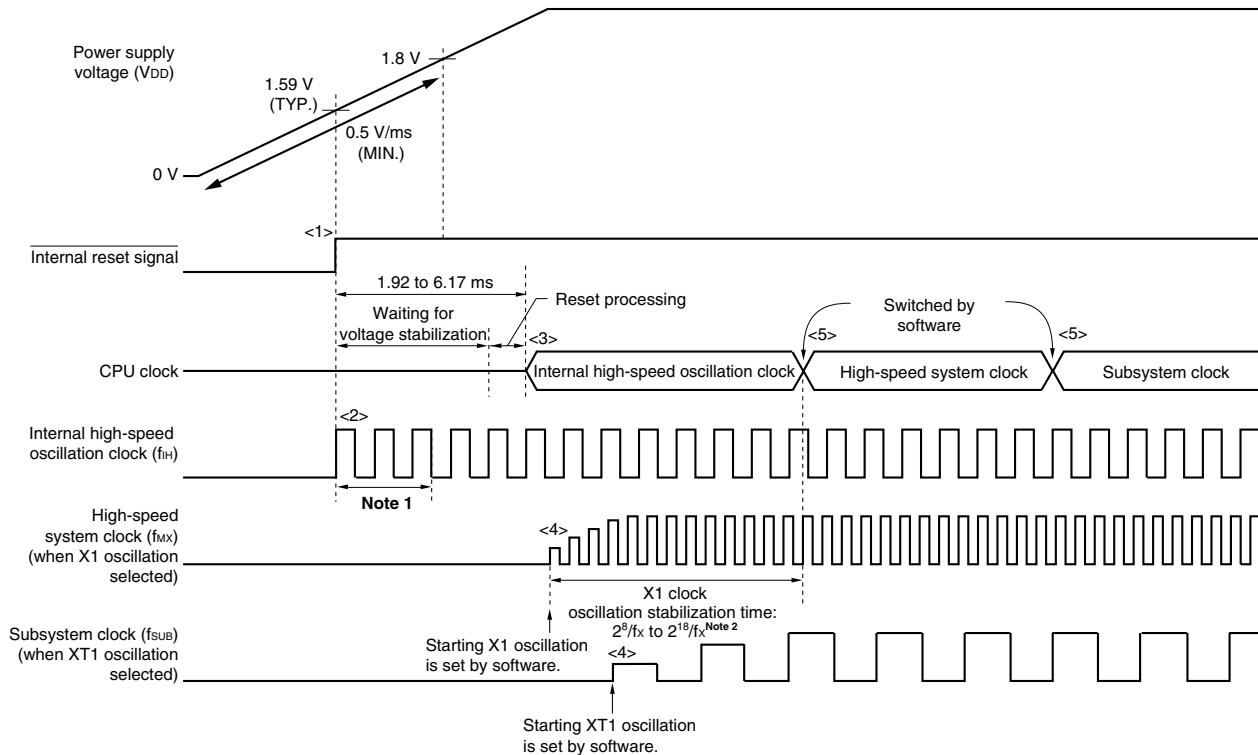
When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the internal high-speed oscillation clock, so the device can be started by the internal high-speed oscillation clock after a reset release. As a result, reset sources can be detected by software and the minimum amount of safety processing can be done during anomalies to ensure that the system terminates safely.

### (2) Improvement of performance

Because the CPU can be started without waiting for the X1 clock oscillation stabilization time, the total performance can be improved.

When the power supply voltage is turned on, the clock generator operation is shown in Figure 6-13 and Figure 6-14.

**Figure 6-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1))**



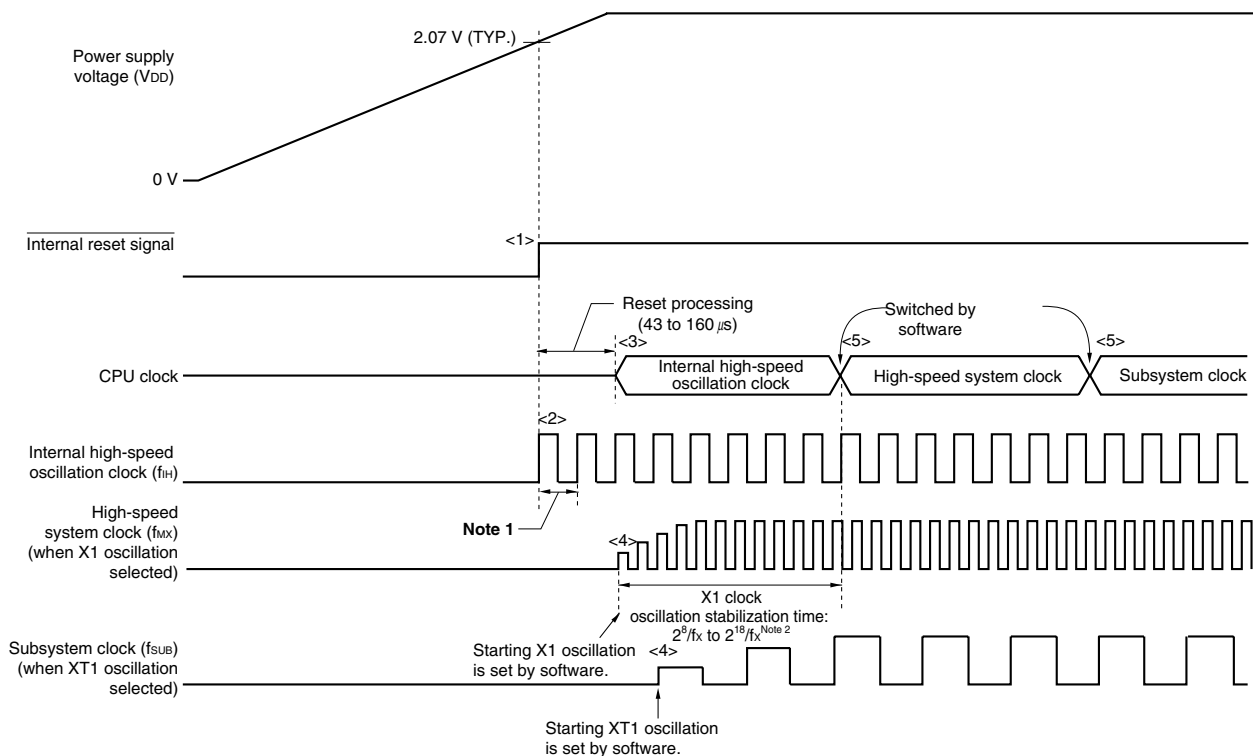
- <1> When the power is turned on, an internal reset signal is generated by the power-on-clear (POC) circuit.
- <2> When the power supply voltage exceeds 1.59 V (TYP.), the reset is released and the internal high-speed oscillator automatically starts oscillation.
- <3> When the power supply voltage rises with a slope of 0.5 V/ms (MIN.), the CPU starts operation on the internal high-speed oscillation clock after the reset is released and after the stabilization times for the voltage of the power supply and regulator have elapsed, and then reset processing is performed.
- <4> Set the start of oscillation of the X1 or XT1 clock via software (see (1) in 6.6.1 Example of controlling high-speed system clock and (1) in 6.6.3 Example of controlling subsystem clock).
- <5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see (3) in 6.6.1 Example of controlling high-speed system clock and (2) in 6.6.3 Example of controlling subsystem clock).

- Notes**
1. The internal voltage stabilization time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.
  2. When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed system clock (X1 oscillation), set the oscillation stabilization time when releasing STOP mode using the oscillation stabilization time select register (OSTS).

- Cautions**
1. If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 1.8 V, input a low level to the  $\overline{\text{RESET}}$  pin from power application until the voltage reaches 1.8 V, or set the LVI default start function stopped by using the option byte (LVIOFF = 0) (see Figure 6-14). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 6-13 after reset release by the  $\overline{\text{RESET}}$  pin.
  2. It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.

**Remark** While the microcontroller is operating, a clock that is not used as the CPU clock can be stopped via software settings. The internal high-speed oscillation clock and high-speed system clock can be stopped by executing the STOP instruction (see (4) in 6.6.1 Example of controlling high-speed system clock, (3) in 6.6.2 Example of controlling internal high-speed oscillation clock, and (3) in 6.6.3 Example of controlling subsystem clock).

**Figure 6-14. Clock Generator Operation When Power Supply Voltage Is Turned On  
(When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0))**



- <1> When the power is turned on, an internal reset signal is generated by the low-voltage detector (LVI).
- <2> When the power supply voltage exceeds 2.07 V (TYP.), the reset is released and the internal high-speed oscillator automatically starts oscillation.
- <3> After the reset is released and reset processing is performed, the CPU starts operation on the internal high-speed oscillation clock.
- <4> Set the start of oscillation of the X1 or XT1 clock via software (see (1) in 6.6.1 Example of controlling high-speed system clock and (1) in 6.6.3 Example of controlling subsystem clock).
- <5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see (3) in 6.6.1 Example of controlling high-speed system clock and (2) in 6.6.3 Example of controlling subsystem clock).

- Notes**
1. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.
  2. When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed system clock (X1 oscillation), set the oscillation stabilization time when releasing STOP mode using the oscillation stabilization time select register (OSTS).

- Cautions**
1. **A voltage oscillation stabilization time is required after the supply voltage reaches 1.59 V (TYP.). If the supply voltage rises from 1.59 V (TYP.) to 2.07 V (TYP.) within the power supply oscillation stabilization time, the power supply oscillation stabilization time is automatically generated before reset processing.**
  2. **It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.**

**Remark** While the microcontroller is operating, a clock that is not used as the CPU clock can be stopped via software settings. The internal high-speed oscillation clock and high-speed system clock can be stopped by executing the STOP instruction (see **(4)** in **6.6.1 Example of controlling high-speed system clock**, **(3)** in **6.6.2 Example of controlling internal high-speed oscillation clock**, and **(3)** in **6.6.3 Example of controlling subsystem clock**).

## 6.6 Controlling Clock

### 6.6.1 Example of controlling high-speed system clock

The following two types of high-speed system clocks are available.

- X1 clock: Crystal/ceramic resonator is connected to the X1 and X2 pins.
- External main system clock: External clock is input to the EXCLK pin.

When the high-speed system clock is not used, the X1/P121 and X2/EXCLK/P122 pins can be used as input port pins.

**Caution** The X1/P121 and X2/EXCLK/P122 pins are in the input port mode after a reset release.

The following describes examples of setting procedures for the following cases.

- (1) When oscillating X1 clock
- (2) When using external main system clock
- (3) When using high-speed system clock as CPU/peripheral hardware clock
- (4) When stopping high-speed system clock

#### (1) Example of setting procedure when oscillating the X1 clock

<1> Setting P121/X1 and P122/X2/EXCLK pins and setting oscillation frequency (CMC register)

- $2 \text{ MHz} \leq f_x \leq 10 \text{ MHz}$

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
0	1	0	0/1	0	0	0	0

- $10 \text{ MHz} < f_x \leq 20 \text{ MHz}$

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
0	1	0	0/1	0	0	0	1

**Remarks 1.**  $f_x$ : X1 clock oscillation frequency

2. For setting of the P123/XT1 and P124/XT2 pins, see **6.6.3 Example of controlling subsystem clock.**

<2> Controlling oscillation of X1 clock (CSC register)

If MSTOP is cleared to 0, the X1 oscillator starts oscillating.

<3> Waiting for the stabilization of the oscillation of X1 clock

Check the OSTC register and wait for the necessary time.

During the wait time, other software processing can be executed with the internal high-speed oscillation clock.

**Cautions 1.** The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction.

Therefore, it is necessary to also set the value of the OSCSELS bit at the same time. For OSCSELS bit, see **6.6.3 Example of controlling subsystem clock.**

2. Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)** and **CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)**).

(2) Example of setting procedure when using the external main system clock

<1> Setting P121/X1 and P122/X2/EXCLK pins (CMC register)

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
1	1	0	0/1	0	0	0	×

Remarks 1. ×: don't care

2. For setting of the P123/XT1 and P124/XT2 pins, see 6.6.3 (1) Example of setting procedure when oscillating the subsystem clock.

<2> Controlling external main system clock input (CSC register)

When MSTOP is cleared to 0, the input of the external main system clock is enabled.

Cautions 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction.

Therefore, it is necessary to also set the value of the OSCSELS bits at the same time. For OSCSELS bits, see 6.6.3 Example of controlling subsystem clock.

2. Set the external main system clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)).

(3) Example of setting procedure when using high-speed system clock as CPU/peripheral hardware clock

<1> Setting high-speed system clock oscillation<sup>Note</sup>

(See 6.6.1 (1) Example of setting procedure when oscillating the X1 clock and (2) Example of setting procedure when using the external main system clock.)

Note The setting of <1> is not necessary when high-speed system clock is already operating.

<2> Setting the high-speed system clock as the source clock of the CPU/peripheral hardware clock and setting the division ratio of the set clock (CKC register)

MCM0	MDIV2	MDIV1	MDIV0	Selection of CPU/Peripheral Hardware Clock (f <sub>CLK</sub> )
1	0	0	0	f <sub>MX</sub>
	0	0	1	f <sub>MX</sub> /2
	0	1	0	f <sub>MX</sub> /2 <sup>2</sup>
	0	1	1	f <sub>MX</sub> /2 <sup>3</sup>
	1	0	0	f <sub>MX</sub> /2 <sup>4</sup>
	1	0	1	f <sub>MX</sub> /2 <sup>5</sup> <sup>Note</sup>

Note Setting is prohibited when f<sub>MX</sub> < 4 MHz.

<3> If some peripheral hardware macros are not used, supply of the input clock to each hardware macro can be stopped.

(PER0 register)

RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
-------	-------	-------	--------	--------	--------	--------	--------

(PER1 register)

0	0	0	0	0	0	0	EXBEN
---	---	---	---	---	---	---	-------

xxxEN	Input clock control
0	Stops input clock supply.
1	Supplies input clock.

**Caution** Be sure to clear bits 1 to 7 of PER1 to 0.

**Remark**

- RTCEN: Control of the real-time counter input clock
- DACEN: Control of the D/A converter input clock
- ADCEN: Control of the A/D converter input clock
- IIC0EN: Control of the serial interface IIC0 input clock
- SAU1EN: Control of the serial array unit 1 input clock
- SAU0EN: Control of the serial array unit 0 input clock
- TAU1EN: Control of the timer array unit 1 input clock
- TAU0EN: Control of the timer array unit 0 input clock
- EXBEN: Control of the external bus interface input clock

#### (4) Example of setting procedure when stopping the high-speed system clock

The high-speed system clock can be stopped (disabling clock input if the external clock is used) in the following two ways.

- Executing the STOP instruction
- Setting MSTOP to 1

##### (a) To execute a STOP instruction

<1> Setting to stop peripheral hardware

Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 19 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after STOP mode is released

If the X1 clock oscillates before the STOP mode is entered, set the value of the OSTS register before executing the STOP instruction.

<3> Executing the STOP instruction

When the STOP instruction is executed, the system is placed in the STOP mode and X1 oscillation is stopped (the input of the external clock is disabled).

**(b) To stop X1 oscillation (disabling external clock input) by setting MSTOP to 1**

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the high-speed system clock.

When CLS = 0 and MCS = 1, the high-speed system clock is supplied to the CPU, so change the CPU clock to the subsystem clock or internal high-speed oscillation clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Setting of X1 clock oscillation stabilization time after restart of X1 clock oscillation<sup>Note</sup>

Prior to setting "1" to MSTOP, set the OSTS register to a value greater than the count value to be confirmed with the OSTS register after X1 clock oscillation is restarted.

<3> Stopping the high-speed system clock (CSC register)

When MSTOP is set to 1, X1 oscillation is stopped (the input of the external clock is disabled).

**Note** This setting is required to resume the X1 clock oscillation when the high-speed system clock is in the X1 oscillation mode.

This setting is not required in the external clock input mode.

**Caution** Be sure to confirm that MCS = 0 or CLS = 1 when setting MSTOP to 1. In addition, stop peripheral hardware that is operating on the high-speed system clock.

**6.6.2 Example of controlling internal high-speed oscillation clock**

The following describes examples of clock setting procedures for the following cases.

- (1) When restarting oscillation of the internal high-speed oscillation clock
- (2) When using internal high-speed oscillation clock as CPU/peripheral hardware clock
- (3) When stopping the internal high-speed oscillation clock

**(1) Example of setting procedure when restarting oscillation of the internal high-speed oscillation clock<sup>Note</sup>**

<1> Setting restart of oscillation of the internal high-speed oscillation clock (CSC register)

When HIOSTOP is cleared to 0, the internal high-speed oscillation clock restarts oscillation.

**Note** After a reset release, the internal high-speed oscillator automatically starts oscillating and the internal high-speed oscillation clock is selected as the CPU/peripheral hardware clock.

**(2) Example of setting procedure when using internal high-speed oscillation clock as CPU/peripheral hardware clock**

<1> Restarting oscillation of the internal high-speed oscillation clock<sup>Note</sup>

(See 6.6.2 (1) Example of setting procedure when restarting internal high-speed oscillation clock).

**Note** The setting of <1> is not necessary when the internal high-speed oscillation clock is operating.



- <2> Setting the internal high-speed oscillation clock as the source clock of the CPU/peripheral hardware clock and setting the division ratio of the set clock (CKC register)

MCM0	MDIV2	MDIV1	MDIV0	Selection of CPU/Peripheral Hardware Clock (f <sub>CLK</sub> )
0	0	0	0	f <sub>IH</sub>
	0	0	1	f <sub>IH</sub> /2
	0	1	0	f <sub>IH</sub> /2 <sup>2</sup>
	0	1	1	f <sub>IH</sub> /2 <sup>3</sup>
	1	0	0	f <sub>IH</sub> /2 <sup>4</sup>
	1	0	1	f <sub>IH</sub> /2 <sup>5</sup>

**Caution** If switching the CPU/peripheral hardware clock from the high-speed system clock to the internal high-speed oscillation clock after restarting the internal high-speed oscillation clock, do so after 10 μs or more have elapsed.

If the switching is made immediately after the internal high-speed oscillation clock is restarted, the accuracy of the internal high-speed oscillation cannot be guaranteed for 10 μs.

### (3) Example of setting procedure when stopping the internal high-speed oscillation clock

The internal high-speed oscillation clock can be stopped in the following two ways.

- Executing the STOP instruction
- Setting HIOSTOP to 1

#### (a) To execute a STOP instruction

- <1> Setting of peripheral hardware

Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 19 STANDBY FUNCTION**).

- <2> Setting the X1 clock oscillation stabilization time after STOP mode is released

If the X1 clock oscillates before the STOP mode is entered, set the value of the OSTS register before executing the STOP instruction.

- <3> Executing the STOP instruction

When the STOP instruction is executed, the system is placed in the STOP mode and internal high-speed oscillation clock is stopped.

#### (b) To stop internal high-speed oscillation clock by setting HIOSTOP to 1

- <1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the internal high-speed oscillation clock.

When CLS = 0 and MCS = 0, the internal high-speed oscillation clock is supplied to the CPU, so change the CPU clock to the high-speed system clock or subsystem clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

- <2> Stopping the internal high-speed oscillation clock (CSC register)  
When HIOSTOP is set to 1, internal high-speed oscillation clock is stopped.

**Caution** Be sure to confirm that MCS = 1 or CLS = 1 when setting HIOSTOP to 1. In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock.

**6.6.3 Example of controlling subsystem clock**

The subsystem clock can be oscillated by connecting a crystal resonator to the XT1 and XT2 pins. When the subsystem clock is not used, the XT1/P123 and XT2/P124 pins can be used as input port pins.

**Caution** The XT1/P123 and XT2/P124 pins are in the input port mode after a reset release.

The following describes examples of setting procedures for the following cases.

- (1) When oscillating subsystem clock
- (2) When using subsystem clock as CPU clock
- (3) When stopping subsystem clock

**Caution** When the subsystem clock is used as the CPU clock, the subsystem clock is also supplied to the peripheral hardware (except the real-time counter, clock output/buzzer output, and watchdog timer). At this time, the operations of the A/D converter and IIC0 are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS).

**(1) Example of setting procedure when oscillating the subsystem clock**

- <1> Setting P123/XT1 and P124/XT2 pins (CMC register)

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
0/1	0/1	0	1	0	0	0	0/1

**Remark** For setting of the P121/X1 and P122/X2 pins, see 6.6.1 Example of controlling high-speed system clock.

- <2> Controlling oscillation of subsystem clock (CSC register)  
If XTSTOP is cleared to 0, the XT1 oscillator starts oscillating.
- <3> Waiting for the stabilization of the subsystem clock oscillation  
Wait for the oscillation stabilization time of the subsystem clock by software, using a timer function.

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

Therefore, it is necessary to also set the value of the EXCLK and OSCSEL bits at the same time. For EXCLK and OSCSEL bits, see 6.6.1 (1) Example of setting procedure when oscillating the X1 clock or 6.6.1 (2) Example of setting procedure when using the external main system clock.

**(2) Example of setting procedure when using the subsystem clock as the CPU clock**

<1> Setting subsystem clock oscillation<sup>Note</sup>

(See 6.6.3 (1) Example of setting procedure when oscillating the subsystem clock.)

**Note** The setting of <1> is not necessary when while the subsystem clock is operating.

<2> Setting the subsystem clock as the source clock of the CPU clock (CKC register)

CSS	Selection of CPU/Peripheral Hardware Clock ( $f_{CLK}$ )
1	$f_{SUB}/2$

**Caution** When the subsystem clock is used as the CPU clock, the subsystem clock is also supplied to the peripheral hardware (except the real-time counter, clock output/buzzer output, and watchdog timer). At this time, the operations of the A/D converter and IIC0 are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS).

**(3) Example of setting procedure when stopping the subsystem clock**

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the subsystem clock.

When CLS = 1, the subsystem clock is supplied to the CPU, so change the CPU clock to the internal high-speed oscillation clock or high-speed system clock (See **Figure 6-15 CPU Clock Status Transition Diagram** or **Table 6-5 Changing CPU Clock** for the conditions to change the subsystem clock to another clock).

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the subsystem clock (CSC register)

When XTSTOP is set to 1, subsystem clock is stopped.

**Cautions** 1. Be sure to confirm that CLS = 0 when setting XTSTOP to 1. In addition, stop the peripheral hardware if it is operating on the subsystem clock.

2. The subsystem clock oscillation cannot be stopped using the STOP instruction.

#### 6.6.4 Example of controlling internal low-speed oscillation clock

The internal low-speed oscillation clock cannot be used as the CPU clock. Used only as the watchdog timer clock.

The internal low-speed oscillator automatically starts oscillation after a reset release, and the watchdog timer is driven (240 kHz (TYP.)) if the watchdog timer operation is enabled by the option byte.

The internal low-speed oscillator continues oscillation except when the watchdog timer stops. When the watchdog timer operates, the internal low-speed oscillation clock does not stop even in case of a program loop.

##### (1) Example of setting procedure when stopping the internal low-speed oscillation clock

The internal low-speed oscillation clock can be stopped in the following two ways.

- Stop the watchdog timer in the HALT/STOP mode by the option byte (bit 0 (WDSTBYON) of 000C0H = 0), and execute the HALT or STOP instruction.
- Stop the watchdog timer by the option byte (bit 4 (WDTON) of 000C0H = 0).

##### (2) Example of setting procedure when restarting oscillation of the internal low-speed oscillation clock

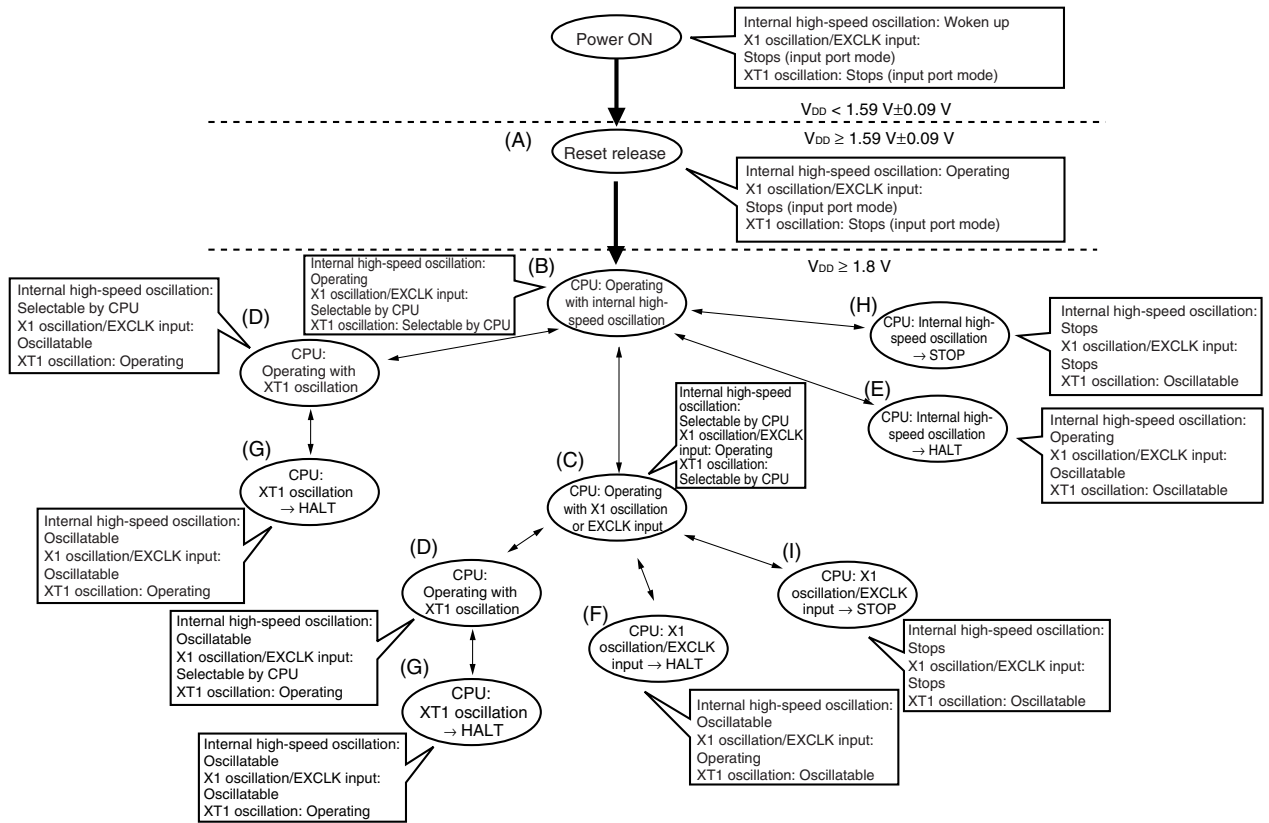
The internal low-speed oscillation clock can be restarted as follows.

- Release the HALT or STOP mode  
(only when the watchdog timer is stopped in the HALT/STOP mode by the option byte (bit 0 (WDSTBYON) of 000C0H) = 0) and when the watchdog timer is stopped as a result of execution of the HALT or STOP instruction).

6.6.5 CPU clock status transition diagram

Figure 6-15 shows the CPU clock status transition diagram of this product.

Figure 6-15. CPU Clock Status Transition Diagram



**Remark** If the low-power-supply detector (LVI) is set to ON by default by the option bytes, the reset will not be released until the power supply voltage ( $V_{DD}$ ) exceeds  $2.07 V \pm 0.2 V$ . After the reset operation, the status will shift to (B) in the above figure.

Table 6-4 shows transition of the CPU clock and examples of setting the SFR registers.

**Table 6-4. CPU Clock Transition and SFR Register Setting Examples (1/4)**

**(1) CPU operating with internal high-speed oscillation clock (B) after reset release (A)**

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

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

(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	CMC Register <sup>Note 1</sup>			CSC Register	OSMC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH	MSTOP	FSEL		MCM0
(A) → (B) → (C) (X1 clock: 2 MHz ≤ f <sub>x</sub> ≤ 10 MHz)	0	1	0	0	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < f <sub>x</sub> ≤ 20 MHz)	0	1	1	0	1 <sup>Note 2</sup>	Must be checked	1
<R> (A) → (B) → (C) (external main clock)	1	1	×	0	0/1	Must not be checked	1

**Notes 1.** The CMC and OSMC registers can be written only once by an 8-bit memory manipulation instruction after reset release.

**2.** FSEL = 1 when f<sub>CLK</sub> > 10 MHz

If a divided clock is selected and f<sub>CLK</sub> ≤ 10 MHz, use with FSEL = 0 is possible even if f<sub>x</sub> > 10 MHz.

**Caution** Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)).

<R> **Remark** ×: don't care

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

(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	CMC Register <sup>Note</sup>	CSC Register	Waiting for Oscillation Stabilization	CKC Register
	OSCSELS	XTSTOP		CSS
(A) → (B) → (D)	1	0	Necessary	1

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

**Remark** (A) to (I) in Table 6-4 correspond to (A) to (I) in Figure 6-15.

Table 6-4. CPU Clock Transition and SFR Register Setting Examples (2/4)

(4) CPU clock changing from internal high-speed oscillation clock (B) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	CMC Register <sup>Note 1</sup>			OSTS Register	CSC Register	OSMC Register	OSTC Register	CKC Register
	EXCLK	OSCESEL	AMPH		MSTOP	FSEL		MCM0
(B) → (C) (X1 clock: 2 MHz ≤ f <sub>X</sub> ≤ 10 MHz)	0	1	0	Note 2	0	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < f <sub>X</sub> ≤ 20 MHz)	0	1	1	Note 2	0	1 <sup>Note 3</sup>	Must be checked	1
(B) → (C) (external main clock)	1	1	×	Note 2	0	0/1	Must not be checked	1

Unnecessary if these registers are already set
 Unnecessary if the CPU is operating with the high-speed system clock

<R>

- Notes**
- The CMC and OSMC registers can be changed only once after reset release. This setting is not necessary if it has already been set.
  - Set the oscillation stabilization time as follows.
    - Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTC
  - FSEL = 1 when f<sub>CLK</sub> > 10 MHz  
 If a divided clock is selected and f<sub>CLK</sub> ≤ 10 MHz, use with FSEL = 0 is possible even if f<sub>X</sub> > 10 MHz.

**Caution** Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)).

<R>

**Remark** x: don't care

(5) CPU clock changing from internal high-speed oscillation clock (B) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	CMC Register <sup>Note</sup>	CSC Register	Waiting for Oscillation Stabilization	CKC Register
	OSCELS	XTSTOP		CSS
(B) → (D)	1	0	Necessary	1

Unnecessary if the CPU is operating with the subsystem clock

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

**Remark** (A) to (I) in Table 6-4 correspond to (A) to (I) in Figure 6-15.

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

(6) CPU clock changing from high-speed system clock (C) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	CSC Register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		MCM0
(C) → (B)	0	10 μs	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

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

(Setting sequence of SFR registers) →

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

Unnecessary if the CPU is operating with the subsystem clock

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

(8) CPU clock changing from subsystem clock (D) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	CSC Register	CKC Register	
	HIOSTOP	MCM0	CSS
(D) → (B)	0	0	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

Unnecessary if this register is already set

**Remark** (A) to (I) in Table 6-4 correspond to (A) to (I) in Figure 6-15.



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

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register Status Transition	OSTS Register	CSC Register	OSMC Register	OSTC Register	CKC Register	
		MSTOP	FSEL		MCM0	CSS
(D) → (C) (X1 clock: $2\text{ MHz} \leq f_x \leq 10\text{ MHz}$ )	<b>Note 1</b>	0	0	Must be checked	1	0
(D) → (C) (X1 clock: $10\text{ MHz} < f_x \leq 20\text{ MHz}$ )	<b>Note 1</b>	0	<sup>1</sup> Note 2	Must be checked	1	0
(D) → (C) (external main clock)	<b>Note 1</b>	0	0/1	Must not be checked	1	0

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

Unnecessary if these registers are already set

**Notes 1.** Set the oscillation stabilization time as follows.

- Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS

**2.** FSEL = 1 when  $f_{CLK} > 10\text{ MHz}$

If a divided clock is selected and  $f_{CLK} \leq 10\text{ MHz}$ , use with FSEL = 0 is possible even if  $f_x > 10\text{ MHz}$ .

**Caution** Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)).

(10) • HALT mode (E) set while CPU is operating with internal high-speed oscillation clock (B)

- HALT mode (F) set while CPU is operating with high-speed system clock (C)
- HALT mode (G) set while CPU is operating with subsystem clock (D)

Status Transition	Setting
(B) → (E) (C) → (F) (D) → (G)	Executing HALT instruction

<R> (11) • STOP mode (H) set while CPU is operating with internal high-speed oscillation clock (B)

- STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence) →

Status Transition		Setting		
(B) → (H)		Stopping peripheral functions that cannot operate in STOP mode	–	Executing STOP instruction
(C) → (I)	In X1 oscillation		Sets the OSTS register	
	External clock		–	

**Remark** (A) to (I) in Table 6-4 correspond to (A) to (I) in Figure 6-15.

**6.6.6 Condition before changing CPU clock and processing after changing CPU clock**

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

**Table 6-5. Changing CPU Clock (1/2)**

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Internal high-speed oscillation clock	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	Operating current can be reduced by stopping internal high-speed oscillator (HIOSTOP = 1).
	External main system clock	Enabling input of external clock from EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	Subsystem clock	Stabilization of X1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	
X1 clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator • HIOSTOP = 0	X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	Subsystem clock	Stabilization of XT1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1).
External main system clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator • HIOSTOP = 0	External main system clock input can be disabled (MSTOP = 1).
	X1 clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	Subsystem clock	Stabilization of XT1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1).

Table 6-5. Changing CPU Clock (2/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Subsystem clock <sup>Note</sup>	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator and selection of internal high-speed oscillation clock as main system clock <ul style="list-style-type: none"> <li>• HIOSTOP = 0, MCS = 0</li> </ul>	XT1 oscillation can be stopped (XTSTOP = 1)
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock <ul style="list-style-type: none"> <li>• OSCSEL = 1, EXCLK = 0, MSTOP = 0</li> <li>• After elapse of oscillation stabilization time</li> <li>• MCS = 1</li> </ul>	
	External main system clock	Enabling input of external clock from EXCLK pin and selection of high-speed system clock as main system clock <ul style="list-style-type: none"> <li>• OSCSEL = 1, EXCLK = 1, MSTOP = 0</li> <li>• MCS = 1</li> </ul>	

**Note** When changing the subsystem clock to another clock, the clock must be set back to the clock before setting the subsystem clock. For example, when changing the clock to the X1 clock after having changed the internal high-speed oscillation clock to the subsystem clock, the clock is changed in the order of the subsystem clock, the internal high-speed oscillation clock, and the X1 clock.

**6.6.7 Time required for switchover of CPU clock and main system clock**

By setting bits 0 to 2, 4, and 6 (MDIV0 to MDIV2, MCM0, CSS) of the system clock control register (CKC), the CPU clock can be switched (between the main system clock and the subsystem clock), main system clock can be switched (between the internal high-speed oscillation clock and the high-speed system clock), and the division ratio of the main system clock can be changed.

The actual switchover operation is not performed immediately after rewriting to CKC; operation continues on the pre-switchover clock for several clocks (see Table 6-6 to Table 6-9).

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

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

Internal high-speed oscillation clock

<R>

**Table 6-6. Maximum Time Required for Main System Clock Switchover**

Clock A	Switching directions	Clock B	Type
$f_{MAINC}$	$\longleftrightarrow$ (Changing the division ratio)	$f_{MAINC}$	Type 1 (see Table 6-7)
$f_{IH}$	$\longleftrightarrow$	$f_{MX}$	Type 2 (see Table 6-8)
$f_{MAINC}$	$\longleftrightarrow$	$f_{SUB}/2$	Type 3 (see Table 6-9)

**Table 6-7. Maximum Number of Clocks Required in Type 1**

Set Value Before Switchover	Set Value After Switchover	
	Clock A	Clock B
Clock A		$1 + f_A/f_B$ clock
Clock B	$1 + f_B/f_A$ clock	

<R>

**Table 6-8. Maximum Number of Clocks Required in Type 2**

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

(Remarks are listed on the next page.)

<R>

**Table 6-9. Maximum Number of Clocks Required in Type 3**

Set Value Before Switchover	Set Value After Switchover	
CSS	CSS	
	0 (f <sub>CLK</sub> = f <sub>MAINC</sub> )	1 (f <sub>CLK</sub> = f <sub>SUB</sub> /2)
0 (f <sub>CLK</sub> = f <sub>MAINC</sub> )		1 + 4 f <sub>MAINC</sub> /f <sub>SUB</sub> clock
1 (f <sub>CLK</sub> = f <sub>SUB</sub> /2)	2 + f <sub>SUB</sub> /2f <sub>MAINC</sub> clock	

<R>

**Remarks 1.** f<sub>IH</sub> :Internal high-speed oscillation clock frequency

f<sub>MX</sub> :High-speed system clock frequency

f<sub>MAIN</sub> :Main system clock frequency

f<sub>MAINC</sub> :Main system select clock frequency

f<sub>SUB</sub> :Subsystem clock frequency

f<sub>CLK</sub> :CPU/peripheral hardware clock frequency

2. The number of clocks listed in Table 6-7 to Table 6-9 is the number of CPU clocks before switchover.
3. Calculate the number of clocks in Table 6-7 to Table 6-9 by removing the decimal portion.

**Example** When switching the main system clock from the internal high-speed oscillation clock to the high-speed system clock (@ oscillation with f<sub>IH</sub> = 8 MHz, f<sub>MX</sub> = 10 MHz)

$$1 + f_{IH}/f_{MX} = 1 + 8/10 = 1 + 0.8 = 1.8 \rightarrow 2 \text{ clocks}$$

### 6.6.8 Conditions before clock oscillation is stopped

The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped.

**Table 6-10. Conditions Before the Clock Oscillation Is Stopped and Flag Settings**

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
Internal high-speed oscillation clock	MCS = 1 or CLS = 1 (The CPU is operating on a clock other than the internal high-speed oscillation clock)	HIOSTOP = 1
X1 clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock)	MSTOP = 1
External main system clock		
Subsystem clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock)	XTSTOP = 1

## CHAPTER 7 TIMER ARRAY UNIT

The timer array unit has two units. The timer array unit 0 has eight 16-bit timers and the timer array unit 1 has four 16-bit timers. Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more “channels” can be used to create a high-accuracy timer.

Single-operation Function	Combination-operation Function
<ul style="list-style-type: none"><li>• Interval timer</li><li>• Square wave output</li><li>• External event counter</li><li>• Divider function (channel 0 of unit 0 only)</li><li>• Input pulse interval measurement</li><li>• Measurement of high-/low-level width of input signal</li></ul>	<ul style="list-style-type: none"><li>• PWM output</li><li>• One-shot pulse output</li><li>• Multiple PWM output</li></ul>

Channel 7 of the timer array unit 0 can be used to realize LIN-bus reception processing in combination with UART3 of serial array unit 1.

### 7.1 Functions of Timer Array Unit

The timer array unit has the following functions.

#### 7.1.1 Functions of each channel when it operates independently

Single-operation functions are those functions that can be used for any channel regardless of the operation mode of the other channel (for details, refer to **7.6.1 Overview of single-operation function and combination-operation function**).

##### (1) Interval timer

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

##### (2) Square wave output

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

##### (3) External event counter

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

##### (4) Divider function (channel 0 of unit 0 only)

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

##### (5) Input pulse interval measurement

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

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

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

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)  
mn = 00 to 07, 10 to 13

**7.1.2 Functions of each channel when it operates with another channel**

Combination-operation functions are those functions that are attained by using the master channel (mostly the reference timer that controls cycles) and the slave channels (timers that operate following the master channel) in combination (for details, refer to **7.6.1 Overview of single-operation function and combination-operation function**).

**(1) PWM (Pulse Width Modulator) output**

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

**(2) One-shot pulse output**

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

**(3) Multiple PWM (Pulse Width Modulator) output**

By extending the PWM function and using one master channel and two or more slave channels, up to seven types of PWM signals that have a specific period and a specified duty factor can be generated.

**7.1.3 LIN-bus supporting function (channel 7 only)****(1) Detection of wakeup signal**

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD3) of UART3 and the count value of the timer is captured at the rising edge. In this way, a low-level width can be measured. If the low-level width is greater than a specific value, it is recognized as a wakeup signal.

**(2) Detection of sync break field**

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD3) of UART3 after a wakeup signal is detected, and the count value of the timer is captured at the rising edge. In this way, a low-level width is measured. If the low-level width is greater than a specific value, it is recognized as a sync break field.

**(3) Measurement of pulse width of sync field**

After a sync break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD3) of UART3 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

## 7.2 Configuration of Timer Array Unit

The timer array unit includes the following hardware.

**Table 7-1. Configuration of Timer Array Unit**

Item	Configuration
Timer/counter	Timer counter register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI07, TI10 to TI13 pins, RxD3 pin (for LIN-bus)
Timer output	TO00 to TO07, TO10 to TO13 pins, output controller
Control registers	<p>&lt;Registers of unit setting block&gt;</p> <ul style="list-style-type: none"> <li>• Peripheral enable register 0 (PER0)</li> <li>• Timer clock select register m (TPSm)</li> <li>• Timer channel enable status register m (TEm)</li> <li>• Timer channel start register m (TSM)</li> <li>• Timer channel stop register m (TTm)</li> <li>• Timer input select register m (TISm)</li> <li>• Timer output enable register m (TOEm)</li> <li>• Timer output register m (TOM)</li> <li>• Timer output level register m (TOLm)</li> <li>• Timer output mode register m (TOMm)</li> </ul> <p>&lt;Registers of each channel&gt;</p> <ul style="list-style-type: none"> <li>• Timer mode register mn (TMRmn)</li> <li>• Timer status register mn (TSRmn)</li> <li>• Input switch control register (ISC) (channel 7 only)</li> <li>• Noise filter enable registers 1, 2 (NFEN1, NFEN2)</li> <li>• Port mode registers 0, 1, 3, 4, 13, 14, 16 (PM0, PM1, PM3, PM4, PM13, PM14, PM16)</li> <li>• Port registers 0, 1, 3, 4, 13, 14, 16 (P0, P1, P3, P4, P13, P14, P16)</li> </ul>

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)  
mn = 00 to 07, 10 to 13

Figure 7-1 and Figure 7-2 shows the block diagram.



Figure 7-1. Block Diagram of Timer Array Unit 0

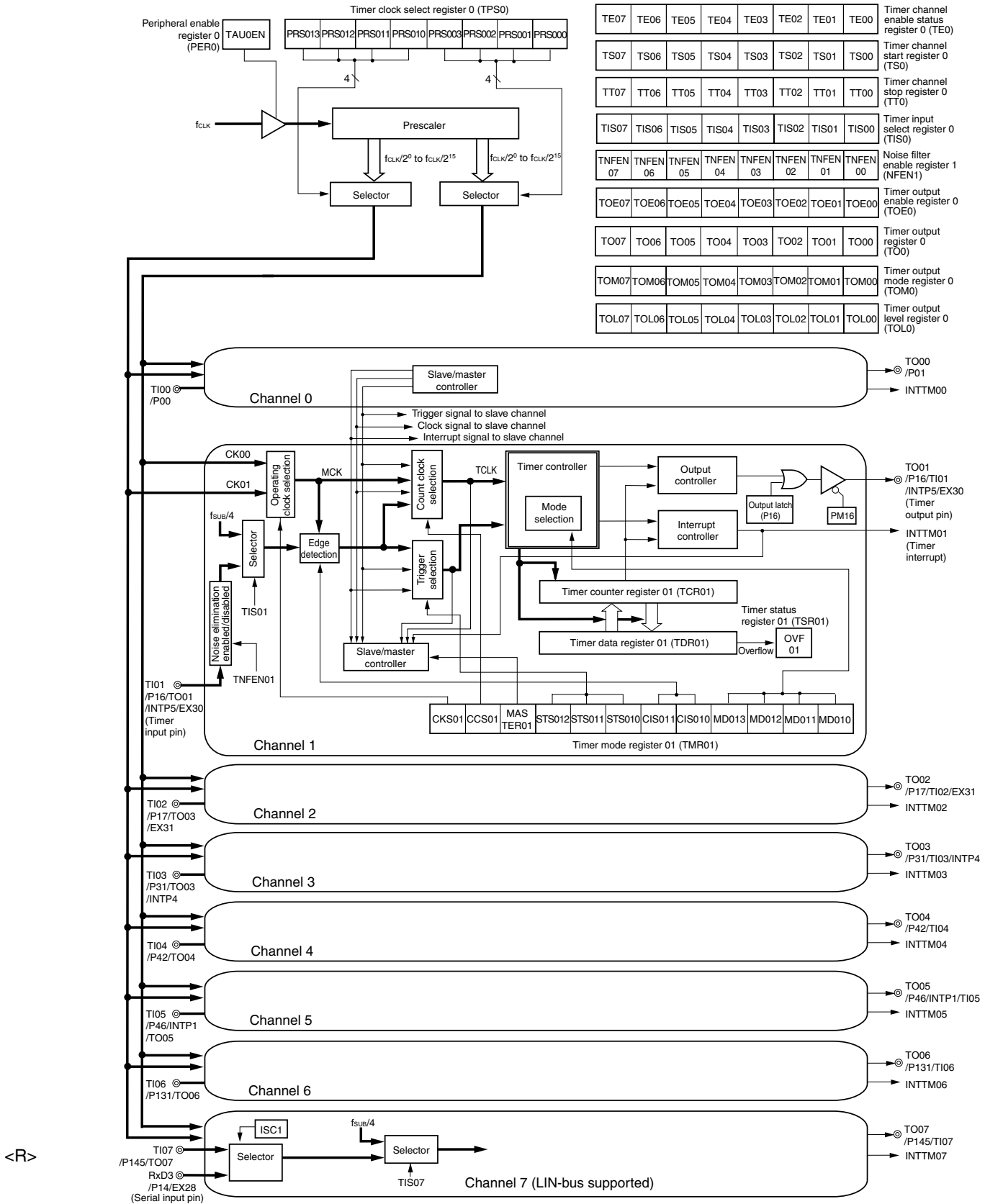
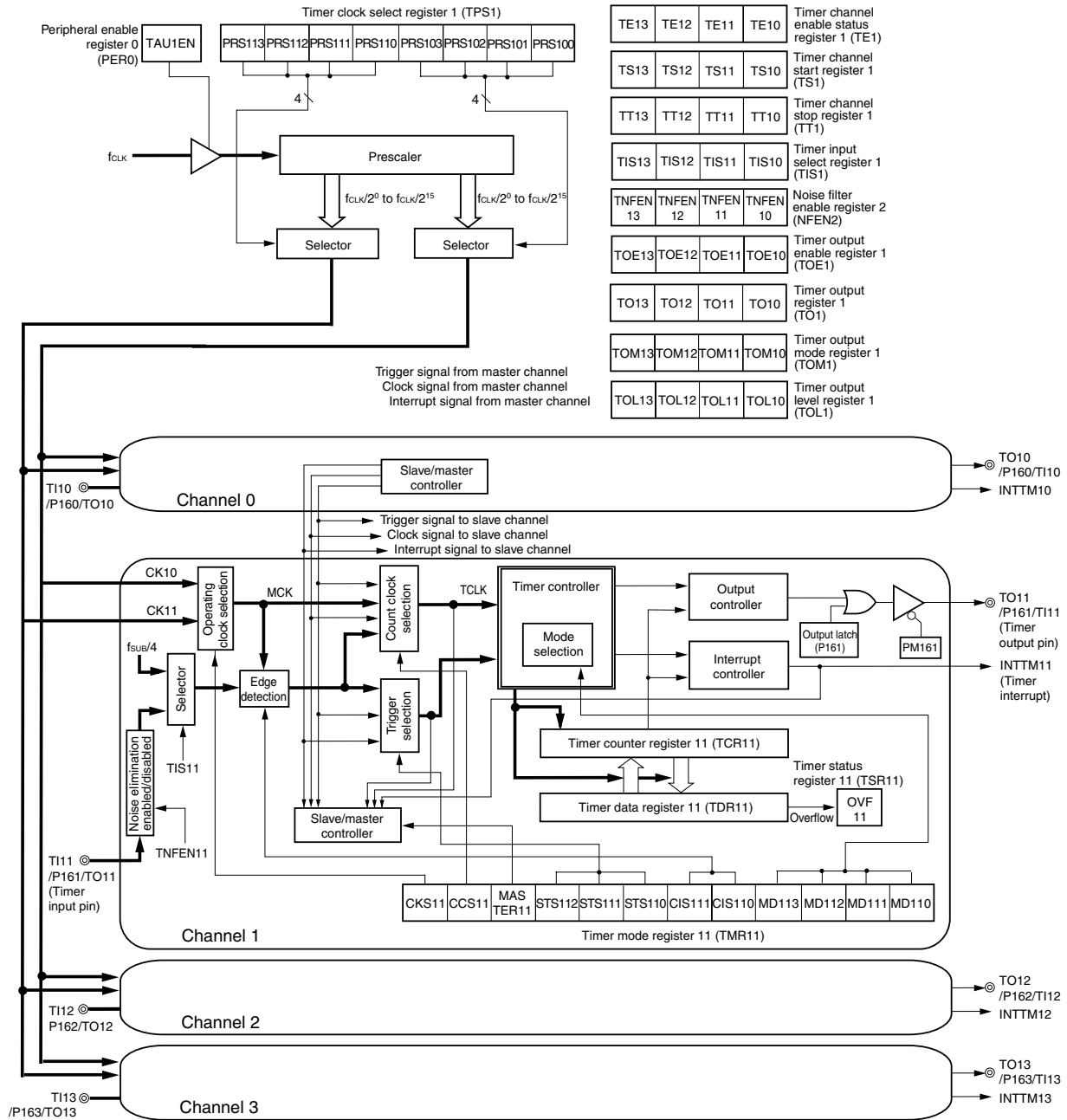


Figure 7-2. Block Diagram of Timer Array Unit 1



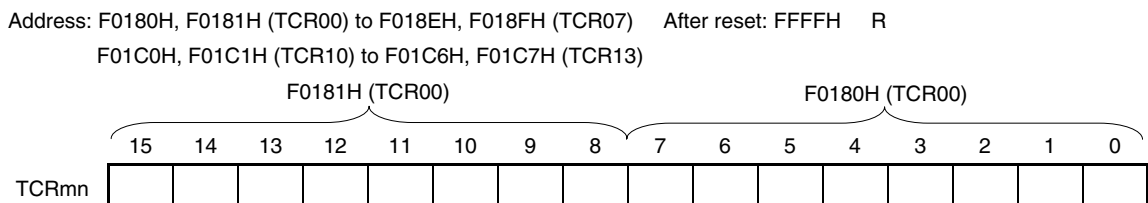
**(1) Timer/counter register mn (TCRmn)**

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

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

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MDmn3 to MDmn0 bits of TMRmn.

**Figure 7-3. Format of Timer/Counter Register mn (TCRmn)**



The count value can be read by reading TCRmn.

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

- When the reset signal is generated
- When the TAU0EN bit (in case of TAU0) or TAU1EN bit (in case of TAU1) of peripheral enable register 0 (PER0) is cleared

When counting of the slave channel has been completed in the PWM output mode

- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

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

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

**Caution** The count value is not captured to TDRmn even when TCRmn is read.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
 mn = 00 to 07, 10 to 13

The TCR<sub>mn</sub> register read value differs as follows according to operation mode changes and the operating status.

**Table 7-2. TCR<sub>mn</sub> Register Read Value in Various Operation Modes**

Operation Mode	Count Mode	TCR <sub>mn</sub> Register Read Value <sup>Note</sup>			
		Operation mode change after reset	Operation mode change after count operation paused (TT <sub>mn</sub> = 1)	Operation restart after count operation paused (TT <sub>mn</sub> = 1)	During start trigger wait status after one count
Interval timer mode	Count down	FFFFH	Undefined	Stop value	–
Capture mode	Count up	0000H	Undefined	Stop value	–
Event counter mode	Count down	FFFFH	Undefined	Stop value	–
One-count mode	Count down	FFFFH	Undefined	Stop value	FFFFH
Capture & one-count mode	Count up	0000H	Undefined	Stop value	Capture value of TDR <sub>mn</sub> register + 1

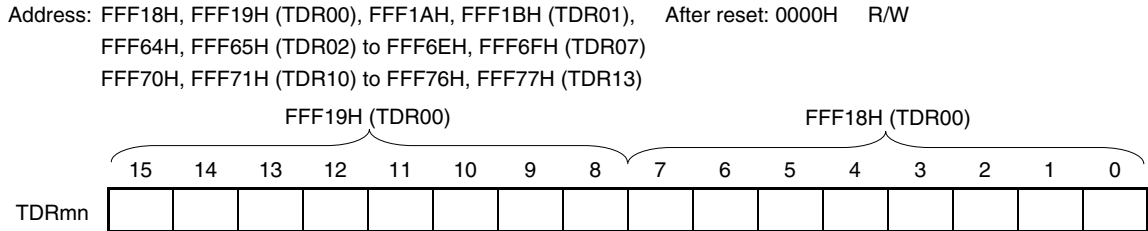
**Note** The read values of the TCR<sub>mn</sub> register when TSM<sub>n</sub> has been set to "1" while TEM<sub>n</sub> = 0 are shown. The read value is held in the TCR<sub>mn</sub> register until the count operation starts.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 13

**(2) Timer data register mn (TDRmn)**

This is a 16-bit register from which a capture function and a compare function can be selected. The capture or compare function can be switched by selecting an operation mode by using the MDmn3 to MDmn0 bits of TMRmn. The value of TDRmn can be changed at any time. This register can be read or written in 16-bit units. Reset signal generation clears this register to 0000H.

**Figure 7-4. Format of Timer Data Register mn (TDRmn)**



**(i) When TDRmn is used as compare register**

Counting down is started from the value set to TDRmn. When the count value reaches 0000H, an interrupt signal (INTTmn) is generated. TDRmn holds its value until it is rewritten.

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

**(ii) When TDRmn is used as capture register**

The count value of TCRmn is captured to TDRmn when the capture trigger is input. A valid edge of the TImn pin can be selected as the capture trigger. This selection is made by TMRmn.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
 mn = 00 to 07, 10 to 13

### 7.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSM)
- Timer channel stop register m (TTm)
- Timer input select register m (TISm)
- Timer output enable register m (TOEm)
- Timer output register m (TOM)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Input switch control register (ISC)
- Noise filter enable registers 1, 2 (NFEN1, NFEN2)
- Port mode registers 0, 1, 3, 4, 13, 14, 16 (PM0, PM1, PM3, PM4, PM13, PM14, PM16)
- Port registers 0, 1, 3, 4, 13, 14, 16 (P0, P1, P3, P4, P13, P14, P16)

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

**(1) Peripheral enable register 0 (PER0)**

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

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

When the timer array unit 1 is used, be sure to set bit 1 (TAU1EN) of this register to 1.

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

Reset signal generation clears this register to 00H.

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

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

TAUmEN	Control of timer array unit m input clock
0	Stops supply of input clock. <ul style="list-style-type: none"> <li>• SFR used by the timer array unit m cannot be written.</li> <li>• The timer array unit m is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the timer array unit m can be read/written.</li> </ul>

**Caution** When setting the timer array unit, be sure to set TAUmEN to 1 first. If TAUmEN = 0, writing to a control register of the timer array unit is ignored, and all read values are default values (except for timer input select register m (TISm), input switch control register (ISC), noise filter enable registers 1, 2 (NFEN1, NFFN2), port mode registers 0, 1, 3, 4, 13, 14, 16 (PM0, PM1, PM3, PM4, PM13, PM14, PM16), and port registers 0, 1, 3, 4, 13, 14, 16 (P0, P1, P3, P4, P13, P14, P16)).

**Remark** m = 0, 1

**(2) Timer clock select register m (TPSm)**

TPSm is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of TPSm, and CKm0 is selected by bits 3 to 0.

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

Rewriting of PRSm00 to PRSm03 bits: Possible only when all the channels set to CKSmn = 0 are in the operation stopped state (TEmn = 0)

Rewriting of PRSm10 to PRSm13 bits: Possible only when all the channels set to CKSmn = 1 are in the operation stopped state (TEmn = 0)

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

The lower 8 bits of TPSm can be set with an 8-bit memory manipulation instruction with TPSmL.

Reset signal generation clears this register to 0000H.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), k = 0, 1  
mn = 00 to 07, 10 to 13



Figure 7-6. Format of Timer Clock Select Register m (TPSm)

Address: F01B6H, F01B7H (TPS0) After reset: 0000H R/W

F01DEH, F01DFH (TPS1)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operation clock (CKmk) <sup>Note</sup>	Selection of operation clock (CKmk) <sup>Note</sup>			
					f <sub>CLK</sub> = 2 MHz	f <sub>CLK</sub> = 5 MHz	f <sub>CLK</sub> = 10 MHz	f <sub>CLK</sub> = 20 MHz
0	0	0	0	f <sub>CLK</sub>	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	f <sub>CLK</sub> /2	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	f <sub>CLK</sub> /2 <sup>2</sup>	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	f <sub>CLK</sub> /2 <sup>5</sup>	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz
0	1	1	0	f <sub>CLK</sub> /2 <sup>6</sup>	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz
0	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	15.62 kHz	39.1 kHz	78.1 kHz	156.2 kHz
1	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	f <sub>CLK</sub> /2 <sup>9</sup>	3.91 kHz	9.76 kHz	19.5 kHz	39.1 kHz
1	0	1	0	f <sub>CLK</sub> /2 <sup>10</sup>	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz
1	0	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	976 Hz	2.44 kHz	4.88 kHz	9.76 kHz
1	1	0	0	f <sub>CLK</sub> /2 <sup>12</sup>	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	0	1	f <sub>CLK</sub> /2 <sup>13</sup>	244 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	0	f <sub>CLK</sub> /2 <sup>14</sup>	122 Hz	305 Hz	610 Hz	1.22 kHz
1	1	1	1	f <sub>CLK</sub> /2 <sup>15</sup>	61 Hz	153 Hz	305 Hz	610 Hz

**Note** When changing the clock selected for f<sub>CLK</sub> (by changing the system clock control register (CKC) value), stop the timer array unit (TT0 = 00FFH, TT1 = 000FH).

**Caution** Be sure to clear bits 15 to 8 to “0”.

- Remarks**
1. f<sub>CLK</sub>: CPU/peripheral hardware clock frequency
  2. m: Unit number (m = 0, 1), k = 0, 1

**(3) Timer mode register mn (TMRmn)**

TMRmn sets an operation mode of channel n. It is used to select an operation clock (MCK), a count clock, whether the timer operates as the master or a slave, a start trigger and a capture trigger, the valid edge of the timer input, and an operation mode (interval, capture, event counter, one-count, or capture & one-count).

Rewriting TMRmn is prohibited when the register is in operation (when TEm = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEm = 1) (for details, see 7.7 **Operation of Timer Array Unit as Independent Channel** and 7.8 **Operation of Plural Channels of Timer Array Unit**).

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

Reset signal generation clears this register to 0000H.

**Figure 7-7. Format of Timer Mode Register mn (TMRmn) (1/3)**

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W  
 F01C8H, F01C9H (TMR10) to F01CEH, F01CFH (TMR13)

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

CKS mn	Selection of operation clock (MCK) of channel n
0	Operation clock CKm0 set by TPSm register
1	Operation clock CKm1 set by TPSm register
Operation clock MCK is used by the edge detector. A count clock (TCLK) is generated depending on the setting of the CCSmn bit.	

CCS mn	Selection of count clock (TCLK) of channel n
0	Operation clock MCK specified by CKSmn bit
1	Valid edge of input signal input from TImn pin/subsystem clock divided by 4 ( $f_{SUB}/4$ )
Count clock TCLK is used for the timer/counter, output controller, and interrupt controller.	

MAS TER mn	Selection of operation in single-operation function or as slave channel in combination-operation function /operation as master channel in combination-operation function of channel n
0	Operates in single-operation function or as slave channel in combination-operation function.
1	Operates as master channel in combination-operation function.
Only the even channel can be set as a master channel (MASTERmn = 1). Be sure to use the odd channel as a slave channel (MASTERmn = 0). Clear MASTERmn to 0 for a channel that is used with the single-operation function.	

**Caution** Be sure to clear bits 14, 13, 5, and 4 to “0”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 13

Figure 7-7. Format of Timer Mode Register mn (TMRmn) (2/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

F01C8H, F01C9H (TMR10) to F01CEH, F01CFH (TMR13)

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

STS mn2	STS mn1	STS mn0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of TImn pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of TImn pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination-operation function).
Other than above			Setting prohibited

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

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

Figure 7-7. Format of Timer Mode Register mn (TMRmn) (3/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

F01C8H, F01C9H (TMR10) to F01CEH, F01CFH (TMR13)

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

MD mn3	MD mn2	MD mn1	MD mn0	Operation mode of channel n	Count operation of TCR	Independent operation
0	0	0	1/0	Interval timer mode	Counting down	Possible
0	1	0	1/0	Capture mode	Counting up	Possible
0	1	1	0	Event counter mode	Counting down	Possible
1	0	0	1/0	One-count mode	Counting down	Impossible
1	1	0	0	Capture & one-count mode	Counting up	Possible
Other than above				Setting prohibited		
The operation of MDmn0 bits varies depending on each operation mode (see table below).						

Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above))	MD mn0	Setting of starting counting and interrupt
<ul style="list-style-type: none"> <li>Interval timer mode (0, 0, 0)</li> <li>Capture mode (0, 1, 0)</li> </ul>	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).
<ul style="list-style-type: none"> <li>Event counter mode (0, 1, 1)</li> </ul>	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
<ul style="list-style-type: none"> <li>One-count mode (1, 0, 0)</li> </ul>	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
	1	Start trigger is valid during counting operation <sup>Note</sup> . At that time, interrupt is also generated.
<ul style="list-style-type: none"> <li>Capture &amp; one-count mode (1, 1, 0)</li> </ul>	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated, either.
Other than above		Setting prohibited

**Note** If the start trigger (TSmn = 1) is issued during operation, the counter is cleared, an interrupt is generated, and recounting is started.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

**(4) Timer status register mn (TSRmn)**

TSRmn indicates the overflow status of the counter of channel n.

TSRmn is valid only in the capture mode (MDmn3 to MDmn1 = 010B) and capture & one-count mode (MDmn3 to MDmn1 = 110B). It will not be set in any other mode. See Table 7-3 for the operation of the OVF bit in each operation mode and set/clear conditions.

TSRmn can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TSRmn can be set with an 8-bit memory manipulation instruction with TSRmnL.

Reset signal generation clears this register to 0000H.

**Figure 7-8. Format of Timer Status Register mn (TSRmn)**

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07) After reset: 0000H R

F01D0H, F01D1H (TSR10) to F01D6H, F01D7H (TSR13)

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

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

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

**Table 7-3. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode**

Timer operation mode	OVF	Set/clear conditions
• Capture mode	clear	When no overflow has occurred upon capturing
• Capture & one-count mode	set	When an overflow has occurred upon capturing
• Interval timer mode	clear	– (Use prohibited, not set and not cleared)
• Event counter mode	set	
• One-count mode		

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

**(5) Timer channel enable status register m (TE<sub>m</sub>)**

TE<sub>m</sub> is used to enable or stop the timer operation of each channel.

When a bit of timer channel start register m (TSM) is set to 1, the corresponding bit of this register is set to 1.

When a bit of timer channel stop register m (TTM) is set to 1, the corresponding bit of this register is cleared to 0.

TE<sub>m</sub> can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TE<sub>m</sub> can be set with a 1-bit or 8-bit memory manipulation instruction with TE<sub>m</sub>L.

Reset signal generation clears this register to 0000H.

**Figure 7-9. Format of Timer Channel Enable Status Register m (TE<sub>m</sub>)**

Address: F01B0H, F01B1H    After reset: 0000H    R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	0	0	0	0	TE07	TE06	TE05	TE04	TE03	TE02	TE01	TE00

Address: F01D8H, F01D9H    After reset: 0000H    R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE1	0	0	0	0	0	0	0	0	0	0	0	0	TE13	TE12	TE11	TE10

TE <sub>mn</sub>	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

**(6) Timer channel start register m (T<sub>Sm</sub>)**

T<sub>Sm</sub> is a trigger register that is used to clear a timer counter (TCR<sub>mn</sub>) and start the counting operation of each channel.

When a bit (T<sub>Smn</sub>) of this register is set to 1, the corresponding bit (TE<sub>mn</sub>) of timer channel enable status register m (TE<sub>m</sub>) is set to 1. T<sub>Smn</sub> is a trigger bit and cleared immediately when TE<sub>mn</sub> = 1.

T<sub>Sm</sub> can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of T<sub>Sm</sub> can be set with a 1-bit or 8-bit memory manipulation instruction with T<sub>SmL</sub>.

Reset signal generation clears this register to 0000H.

**Figure 7-10. Format of Timer Channel Start Register m (T<sub>Sm</sub>)**

Address: F01B2H, F01B3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	0	0	0	0	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

Address: F01DAH, F01DBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1	0	0	0	0	0	0	0	0	0	0	0	0	TS13	TS12	TS11	TS10

TS mn	Operation enable (start) trigger of channel n
0	No trigger operation
1	TE <sub>mn</sub> is set to 1 and the count operation becomes enabled. The TCR <sub>mn</sub> count operation start in the count operation enabled state varies depending on each operation mode (see Table 7-4).

**Caution** Be sure to clear bits 15 to 8 of TS0 and bits 15 to 4 of TS1 to “0”

**Remarks 1.** When the T<sub>Sm</sub> register is read, 0 is always read.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

**Table 7-4. Operations from Count Operation Enabled State to TCR<sub>mn</sub> Count Start (1/2)**

Timer operation mode	Operation when T <sub>Smn</sub> = 1 is set
<ul style="list-style-type: none"> <li>• Interval timer mode</li> </ul>	No operation is carried out from start trigger detection (T <sub>Smn</sub> =1) until count clock generation.  The first count clock loads the value of TDR <sub>mn</sub> to TCR <sub>mn</sub> and the subsequent count clock performs count down operation (see <b>7.3 (6) (a) Start timing in interval timer mode</b> ).
<ul style="list-style-type: none"> <li>• Event counter mode</li> </ul>	Writing 1 to T <sub>Smn</sub> bit loads the value of TDR <sub>mn</sub> to TCR <sub>mn</sub> . The subsequent count clock performs count down operation. The external trigger detection selected by STS <sub>mn2</sub> to STS <sub>mn0</sub> bits in the TMR <sub>mn</sub> register does not start count operation (see <b>7.3 (6) (b) Start timing in event counter mode</b> ).
<ul style="list-style-type: none"> <li>• Capture mode</li> </ul>	No operation is carried out from start trigger detection until count clock generation.  The first count clock loads 0000H to TCR <sub>mn</sub> and the subsequent count clock performs count up operation (see <b>7.3 (6) (c) Start timing in capture mode</b> ).

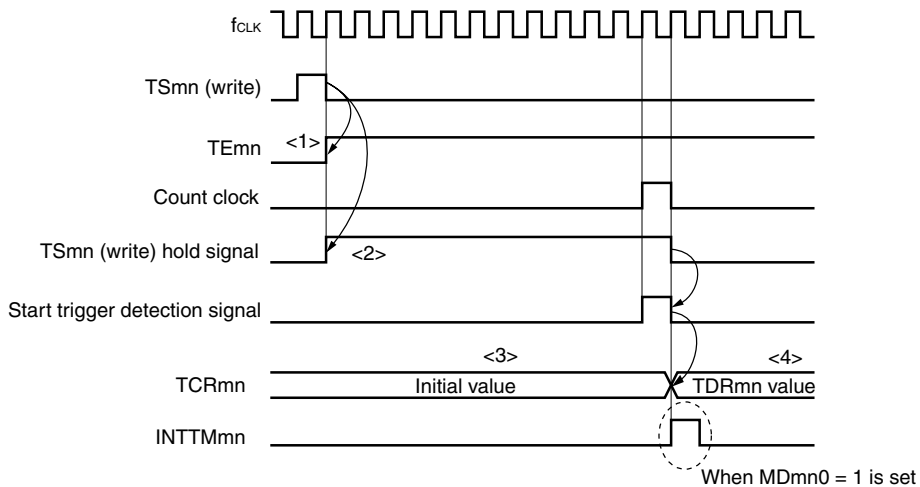
**Table 7-4. Operations from Count Operation Enabled State to TCRmn Count Start (2/2)**

Timer operation mode	Operation when TSmn = 1 is set
<ul style="list-style-type: none"> <li>One-count mode</li> </ul>	When TSmn = 0, writing 1 to TSmn bit sets the start trigger wait state. No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of TDRmn to TCRmn and the subsequent count clock performs count down operation (see 7.3 (6) (d) <b>Start timing in one-count mode</b> ).
<ul style="list-style-type: none"> <li>Capture &amp; one-count mode</li> </ul>	When TSmn = 0, writing 1 to TSmn bit sets the start trigger wait state. No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to TCRmn and the subsequent count clock performs count up operation (see 7.3 (6) (e) <b>Start timing in capture &amp; one-count mode</b> ).

**(a) Start timing in interval timer mode**

- <1> Writing 1 to TSmn sets TE<sub>mn</sub> = 1
- <2> The write data to TSmn is held until count clock generation.
- <3> TCRmn holds the initial value until count clock generation.
- <4> On generation of count clock, the “TDRmn value” is loaded to TCRmn and count starts.

**Figure 7-11. Start Timing (In Interval Timer Mode)**



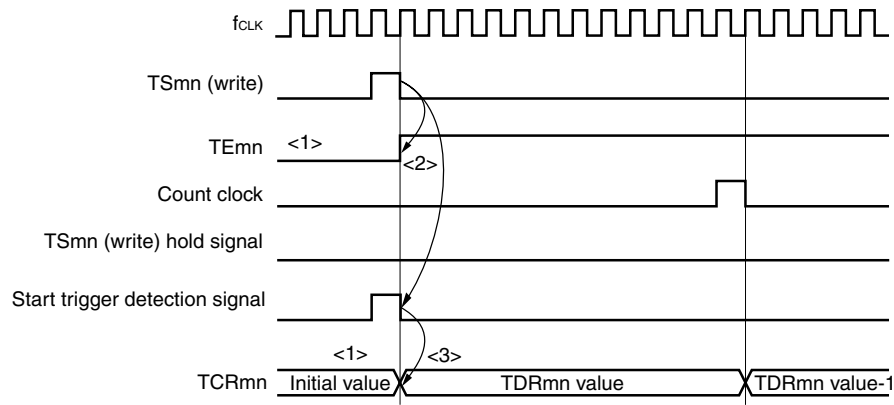
**Caution** In the first cycle operation of count clock after writing TSmn, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.



**(b) Start timing in event counter mode**

- <1> While TEmn is set to 0, TCRmn holds the initial value.
- <2> Writing 1 to TSmn sets 1 to TEmn.
- <3> As soon as 1 has been written to TSmn and 1 has been set to TEmn, the “TDRmn value” is loaded to TCRmn to start counting.
- <4> After that, the TCRmn value is counted down according to the count clock.

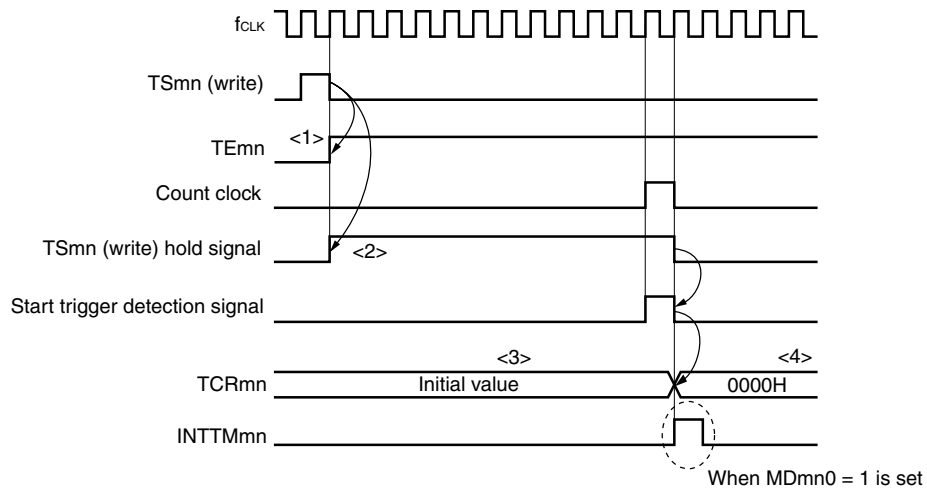
**Figure 7-12. Start Timing (In Event Counter Mode)**



**(c) Start timing in capture mode**

- <1> Writing 1 to TSmn sets TEmn = 1
- <2> The write data to TSmn is held until count clock generation.
- <3> TCRmn holds the initial value until count clock generation.
- <4> On generation of count clock, 0000H is loaded to TCRmn and count starts.

**Figure 7-13. Start Timing (In Capture Mode)**



**Caution** In the first cycle operation of count clock after writing TSmn, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

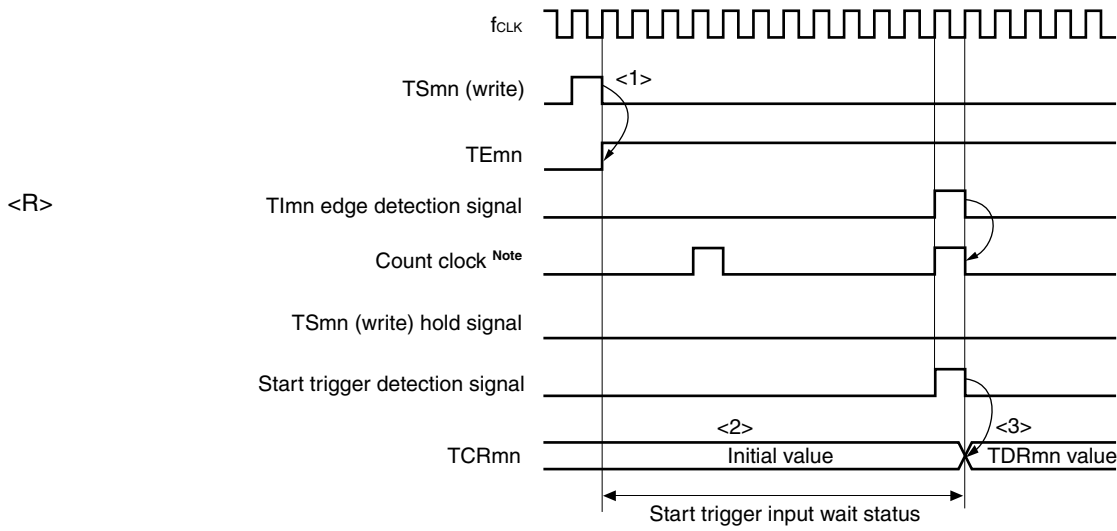
(d) Start timing in one-count mode

<1> Writing 1 to TSmn sets TEMn = 1

<2> Enters the start trigger input wait status, and TCRmn holds the initial value.

<3> On start trigger detection, the “TDRmn value” is loaded to TCRmn and count starts.

Figure 7-14. Start Timing (In One-count Mode)



**Note** When the one-count mode is set, the operation clock (MCK) is selected as count clock (CCSmn = 0).

**Caution** An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TImn is used).

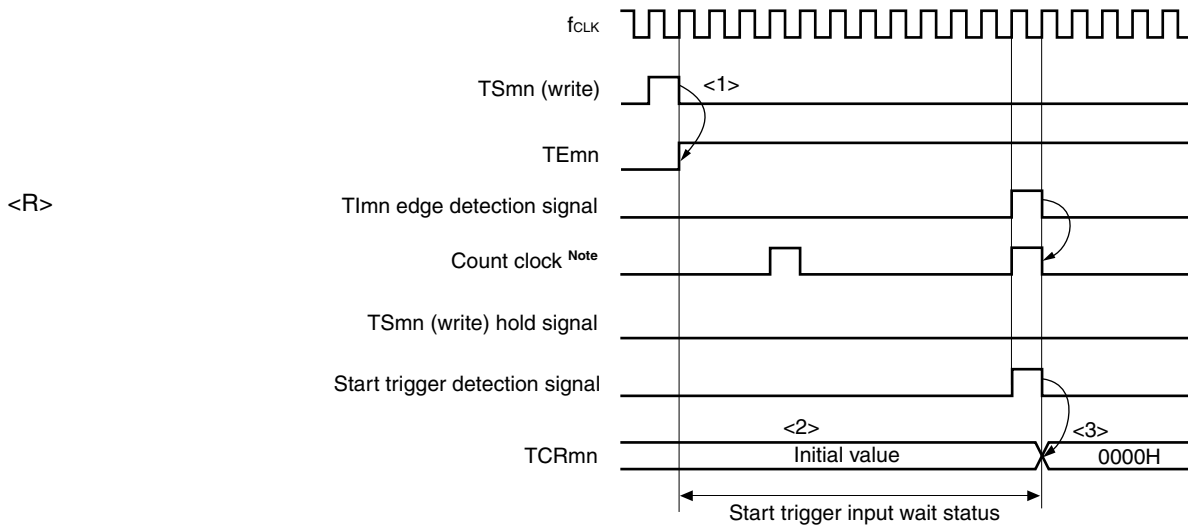
(e) Start timing in capture & one-count mode

<1> Writing 1 to TSmn sets TEMn = 1

<2> Enters the start trigger input wait status, and TCRmn holds the initial value.

<3> On start trigger detection, 0000H is loaded to TCRmn and count starts.

Figure 7-15. Start Timing (In Capture & One-count Mode)



**Note** When the capture & one-count mode is set, the operation clock (MCK) is selected as count clock (CCSmn = 0).

**Caution** An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TImn is used).

**(7) Timer channel stop register m (TTm)**

TTm is a trigger register that is used to clear a timer counter (TCRmn) and start the counting operation of each channel.

When a bit (TTmn) of this register is set to 1, the corresponding bit (TEmn) of timer channel enable status register m (TEM) is cleared to 0. TTmn is a trigger bit and cleared to 0 immediately when TEMn = 0.

TTm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TTm can be set with a 1-bit or 8-bit memory manipulation instruction with TTmL.

Reset signal generation clears this register to 0000H.

**Figure 7-16. Format of Timer Channel Stop Register m (TTm)**

Address: F01B4H, F01B5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0	0	0	0	0	0	0	0	0	TT07	TT06	TT05	TT04	TT03	TT02	TT01	TT00

Address: F01DCH, F01DDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT1	0	0	0	0	0	0	0	0	0	0	0	0	TT13	TT12	TT11	TT10

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	Operation is stopped (stop trigger is generated).

**Caution** Be sure to clear bits 15 to 8 of TT0 and bits 15 to 4 of TT1 to “0”.

- Remarks**
1. When the TTm register is read, 0 is always read.
  2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 13

**(8) Timer input select register m (TISm)**

TISm is used to select whether a signal input to the timer input pin (TImn) or the subsystem clock divided by four ( $f_{SUB}/4$ ) is valid for each channel.

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

Reset signal generation clears this register to 00H.

**Figure 7-17. Format of Timer Input Select Register m (TISm)**

Address: FFF3EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	TIS07	TIS06	TIS05	TIS04	TIS03	TIS02	TIS01	TIS00

Address: FFF3FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS1	0	0	0	0	TIS13	TIS12	TIS11	TIS10

TISmn	Selection of timer input/subsystem clock used with channel n
0	Input signal of timer input pin (TImn)
1	Subsystem clock divided by 4 ( $f_{SUB}/4$ )

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

**(9) Timer output enable register m (TOEm)**

TOEm is used to enable or disable timer output of each channel.

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

TOEm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOEm can be set with a 1-bit or 8-bit memory manipulation instruction with TOEmL.

Reset signal generation clears this register to 0000H.

**Figure 7-18. Format of Timer Output Enable Register m (TOEm)**

Address: F01BAH, F01BBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE0	0	0	0	0	0	0	0	0	TOE 07	TOE 06	TOE 05	TOE 04	TOE 03	TOE 02	TOE 01	TOE 00

Address: F01E2H, F01E3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE1	0	0	0	0	0	0	0	0	0	0	0	0	TOE 13	TOE 12	TOE 11	TOE 10

TOE mn	Timer output enable/disable of channel n
0	The TOMn operation stopped by count operation (timer channel output bit). Writing to the TOMn bit is enabled. The TOMn pin functions as data output, and it outputs the level set to the TOMn bit. The output level of the TOMn pin can be manipulated by software.
1	The TOMn operation enabled by count operation (timer channel output bit). Writing to the TOMn bit is disabled (writing is ignored). The TOMn pin functions as timer output, and the TOEmn is set or reset depending on the timer operation. The TOMn pin outputs the square-wave or PWM depending on the timer operation.

**Caution** Be sure to clear bits 15 to 8 of TOE0 and bits 15 to 4 of TOE1 to “0”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

**(10) Timer output register m (TOm)**

TOm is a buffer register of timer output of each channel.

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

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

To use the P01/TO00, P16/TO01, P17/TO02, P31/TO03, P42/TO04, P46/TO05, P131/TO06, P145/TO07, P160/TO10, P161/TO11, P162/TO12, or P163/TO13 pin as a port function pin, set the corresponding TOmn bit to "0".

TOm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOm can be set with an 8-bit memory manipulation instruction with TOML.

Reset signal generation clears this register to 0000H.

**Figure 7-19. Format of Timer Output Register m (TOm)**

Address: F01B8H, F01B9H    After reset: 0000H    R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	0	0	TO0 7	TO0 6	TO0 5	TO0 4	TO0 3	TO0 2	TO0 1	TO0 0

Address: F01E0H, F01E1H    After reset: 0000H    R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO1	0	0	0	0	0	0	0	0	0	0	0	0	TO1 3	TO1 2	TO1 1	TO1 0

TO mn	Timer output of channel n															
0	Timer output value is "0".															
1	Timer output value is "1".															

**Caution** Be sure to clear bits 15 to 8 of TO0 and bits 15 to 4 of TO1 to "0".

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

**(11) Timer output level register m (TOLm)**

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

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOEmn = 1) in the combination-operation mode (TOMmn = 1). In the toggle mode (TOMmn = 0), this register setting is invalid.

TOLm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOLm can be set with an 8-bit memory manipulation instruction with TOLmL.

Reset signal generation clears this register to 0000H.

**Figure 7-20. Format of Timer Output Level Register m (TOLm)**

Address: F01BCH, F01BDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	0	0	0	0	TOL 07	TOL 06	TOL 05	TOL 04	TOL 03	TOL 02	TOL 01	TOL 00

Address: F01E4H, F01E5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL1	0	0	0	0	0	0	0	0	0	0	0	0	TOL 13	TOL 12	TOL 11	TOL 10

TOL mn	Control of timer output level of channel n														
0	Positive logic output (active-high)														
1	Inverted output (active-low)														

**Caution** Be sure to clear bits 15 to 8 of TOL0 and bits 15 to 4 of TOL1 to “0”.

- Remarks 1.** If the value of this register is rewritten during timer operation, the timer output is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.
- 2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13



**(12) Timer output mode register m (TOMm)**

TOMm is used to control the timer output mode of each channel.

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

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

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

TOMm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOMm can be set with an 8-bit memory manipulation instruction with TOMmL.

Reset signal generation clears this register to 0000H.

**Figure 7-21. Format of Timer Output Mode Register m (TOMm)**

Address: F01BEH, F01BFH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	0	0	TOM 07	TOM 06	TOM 05	TOM 04	TOM 03	TOM 02	TOM 01	TOM 00

Address: F01E6, F01E7H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM1	0	0	0	0	0	0	0	0	0	0	0	0	TOM 13	TOM 12	TOM 11	TOM 10

TOM mn	Control of timer output mode of channel n
0	Toggle mode (to produce toggle output by timer interrupt request signal (INTTMmn))
1	Combination operation mode (output is set by the timer interrupt request signal (INITTMmn) of the master channel, and reset by the timer interrupt request signal (INITTMmp) of the slave channel)

**Caution** Be sure to clear bits 15 to 8 of TOM0 and bits 15 to 4 of TOM1 to “0”.

**Remark** m: Unit number, n: Channel number, p: Slave channel number

When m = 0

n = 0 to 7 (n = 0, 2, 4, 6 for master channel)

n < p ≤ 7 (where p is a consecutive integer greater than n)

When m = 1

n = 0 to 3 (n = 0, 2 for master channel)

n < p ≤ 3 (where p is a consecutive integer greater than n)

**(13) Input switch control register (ISC)**

ISC is used to implement LIN-bus communication operation with channel 7 in association with serial array unit 1.

When bit 1 of this register is set to 1, the input signal of the serial data input pin (RxD3) is selected as a timer input signal.

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

Reset signal generation clears this register to 00H.

**Figure 7-22. Format of Input Switch Control Register (ISC)**

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of RxD3 pin is used as timer input (wakeup signal detection). (to measure the pulse widths of the sync break field and sync field).

<R>

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD3 pin as an external interrupt (wakeup signal detection).

<R>

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

**Remark** When the LIN-bus communication function is used, select the input signal of the RxD3 pin by setting ISC1 to 1 and setting TIS07 (bit 7 of the timer input select register 0 (TIS0)) to 0.

**(14) Noise filter enable registers 1, 2 (NFEN1, NFEN2)**

NFEN1 and NFEN2 are 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 2 clocks is performed with the CPU/peripheral hardware clock (f<sub>CLK</sub>). When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock (f<sub>CLK</sub>).

NFEN1 and NFEN2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-23. Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0061H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN07	Enable/disable using noise filter of TI07/TO07/P145 pin or RxD3/P14/EX26 pin input signal <sup>Note</sup>
0	Noise filter OFF
1	Noise filter ON

TNFEN06	Enable/disable using noise filter of TI06/TO06/P131 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN05	Enable/disable using noise filter of TI05/TO05/P46 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN04	Enable/disable using noise filter of TI04/TO04/P42 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03/TO03/INTP4/P31 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02/TO02/P17 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01/TO01/INTP5/P16 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00/P00 pin input signal
0	Noise filter OFF
1	Noise filter ON

**Note** The applicable pin can be switched by setting ISC1 of the ISC register.

ISC1 = 0: Whether or not to use the noise filter of TI07 pin can be selected.

ISC1 = 1: Whether or not to use the noise filter of RxD3 pin can be selected.

**Figure 7-24. Format of Noise Filter Enable Register 2 (NFEN2)**

Address: F0062H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
NFEN2	0	0	0	0	TNFEN13	TNFEN12	TNFEN11	TNFEN10

TNFEN13	Enable/disable using noise filter of T113/TO13/P163 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN12	Enable/disable using noise filter of T112/TO12/P162 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN11	Enable/disable using noise filter of T111/TO11/P161 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN10	Enable/disable using noise filter of T110/TO10/P160 pin input signal
0	Noise filter OFF
1	Noise filter ON

**(15) Port mode registers 0, 1, 3, 4, 13, 14, 16 (PM0, PM1, PM3, PM4, PM13, PM14, PM16)**

These registers set input/output of ports 0, 1, 3, 4, 13, 14 and 16 in 1-bit units.

When using the P01/TO00, P16/TO01/TI01/INTP5/EX30, P17/TO02/TI02/EX31, P31/TO03/TI03/INTP4, P42/TO04/TI04, P46/TO05/TI05/INTP1, P131/TO06/TI06, P145/TO07/TI07, and P160/TI10/TO10 to P163/TI13/TO13 pins for timer output, set PM01, PM16, PM17, PM31, PM42, PM46, PM131, PM145, and PM160 to PM163 and the output latches of P01, P16, P17, P31, P42, P46, P131, P145, and P160 to P163 to 0.

When using the P00/TI00, P16/TO01/TI01/INTP5/EX30, P17/TO02/TI02/EX31, P31/TO03/TI03/INTP4, P42/TO04/TI04, P46/TO05/TI05/INTP1, P131/TO06/TI06, P145/TO07/TI07, and P160/TI10/TO10 to P163/TO13/TI13 pins for timer input, set PM00, PM16, PM17, PM31, PM42, PM46, PM131, PM145, and PM160 to PM163 to 1. At this time, the output latches of P00, P16, P17, P31, P42, P46, P131, P145, and P160 to P163 may be 0 or 1.

PM0, PM1, PM3, PM4, PM13, PM14, and PM16 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

**Figure 7-25. Format of Port Mode Registers 0, 1, 3, 4, 13, 14, 16 (PM0, PM1, PM3, PM4, PM13, PM14, PM16)**

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

Address: FFF2DH After reset: FEH R/W

Symbol	7	6	5	4	3	2	1	0
PM13	1	1	1	1	1	1	PM131	0

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140

Address: FFF35H After reset: FFH R/W

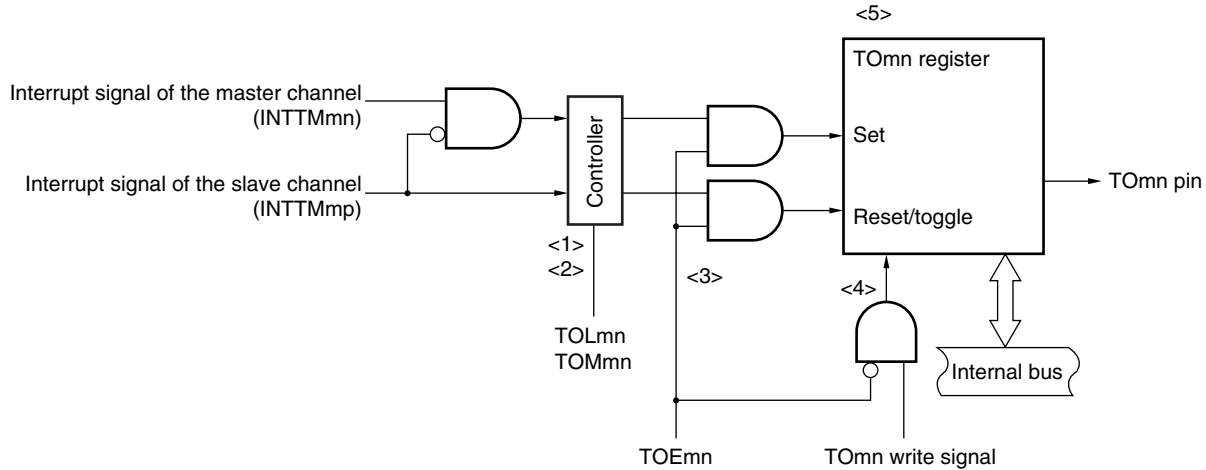
Symbol	7	6	5	4	3	2	1	0
PM16	1	1	1	1	PM163	PM162	PM161	PM160

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 4, 13, 14, 16; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

## 7.4 Channel Output (TOMn pin) Control

### 7.4.1 TOMn pin output circuit configuration

Figure 7-26. Output Circuit Configuration



The following describes the TOMn pin output circuit.

- <1> When TOMmn = 0 (toggle mode), the set value of the TOLmn register is ignored and only INTTMmp (slave channel timer interrupt) is transmitted to the TOMn register.
- <2> When TOMmn = 1 (combination-operation mode), both INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are transmitted to the TOMn register.

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

- When TOLmn = 0: Forward operation (INTTMmn → set, INTTMmp → reset)
- When TOLmn = 1: Reverse operation (INTTMmn → reset, INTTMmp → set)

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

- <3> When TOEmn = 1, INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are transmitted to the TOMn register. Writing to the TOMn register (TOMn write signal) becomes invalid.
  - When TOEmn = 1, the TOMn pin output never changes with signals other than interrupt signals.
  - To initialize the TOMn pin output level, it is necessary to set TOEmn = 0 and to write a value to TOMn.
- <4> When TOEmn = 0, writing to TOMn bit to the target channel (TOMn signal) becomes valid. When TOEmn = 0 neither INTTMmn (master channel timer interrupt) nor INTTMmp (slave channel timer interrupt) is transmitted to TOMn register.
- <5> The TOMn register can always be read, and the TOMn pin output level can be checked.

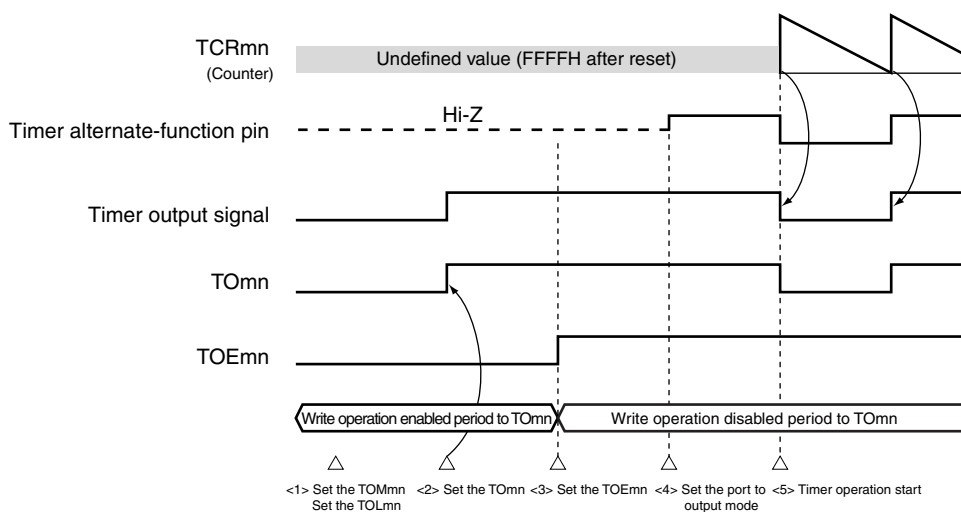
(Remark is given on the next page.)

**Remark** m: Unit number, n: Channel number, p: Slave channel number  
 When m = 0  
     n = 0 to 7 (n = 0, 2, 4, 6 for master channel)  
     n < p ≤ 7 (where p is a consecutive integer greater than n)  
 When m = 1  
     n = 0 to 3 (n = 0, 2 for master channel)  
     n < p ≤ 3 (where p is a consecutive integer greater than n)

**7.4.2 TOMn Pin Output Setting**

The following figure shows the procedure and status transition of TOMn out put pin from initial setting to timer operation start.

**Figure 7-27. Status Transition from Timer Output Setting to Operation Start**



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

- TOMmn bit (0: Toggle mode, 1: Combination-operation mode)
- TOLmn bit (0: Forward output, 1: Reverse output)

<2> The timer output signal is set to the initial status by setting TOMn.

<3> The timer output operation is enabled by writing 1 to TOEmn (writing to TOMn is disabled).

<4> The port I/O setting is set to output (see 7.3 (15) **Port mode registers 0, 1, 3, 4, 13, 14, 16**).

<5> The timer operation is enabled (TSmn = 1).

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
 mn = 00 to 07, 10 to 13

7.4.3 Cautions on Channel Output Operation

(1) Changing values set in registers TOM, TOEm, TOLm, and TOMm during timer operation

Since the timer operations (operations of TCRmn and TDRmn) are independent of the TOMn output circuit and changing the values set in TOM, TOEm, TOLm, and TOMm does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TOMn pin by timer operation, however, set TOM, TOEm, TOLm, and TOMm to the values stated in the register setting example of each operation.

When the values set in TOEm, TOLm, and TOMm (except for TOM) are changed close to the timer interrupt (INTTMmn), the waveform output to the TOMn pin may be different depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMmn) signal generation timing.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

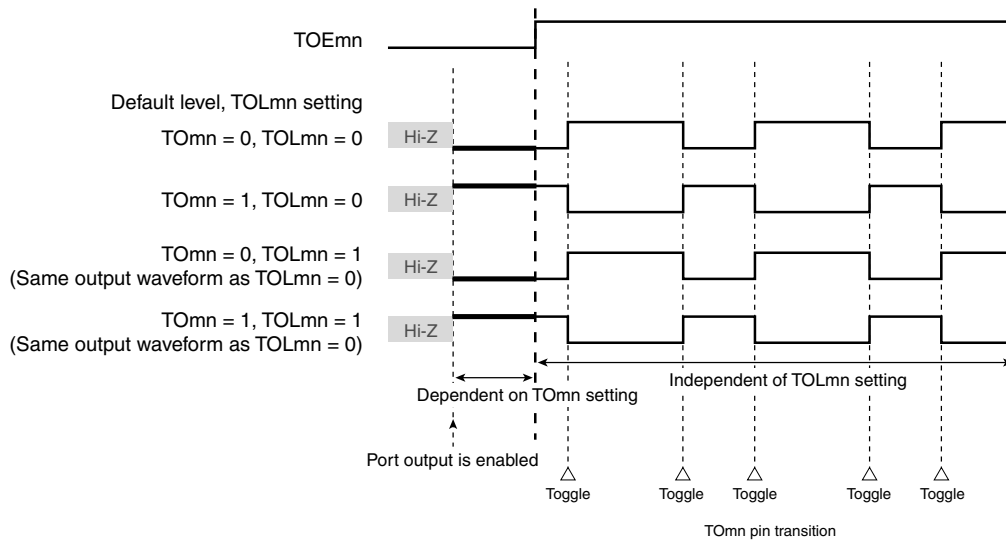
(2) Default level of TOMn pin and output level after timer operation start

The following figure shows the TOMn pin output level transition when writing has been done in the state of TOEmn = 0 before port output is enabled and TOEmn = 1 is set after changing the default level.

(a) When operation starts with TOMmn = 0 setting (toggle output)

The setting of TOLmn is invalid when TOMmn = 0. When the timer operation starts after setting the default level, the toggle signal is generated and the output level of TOMn pin is reversed.

Figure 7-28. TOMn Pin Output Status at Toggle Output (TOMmn = 0)



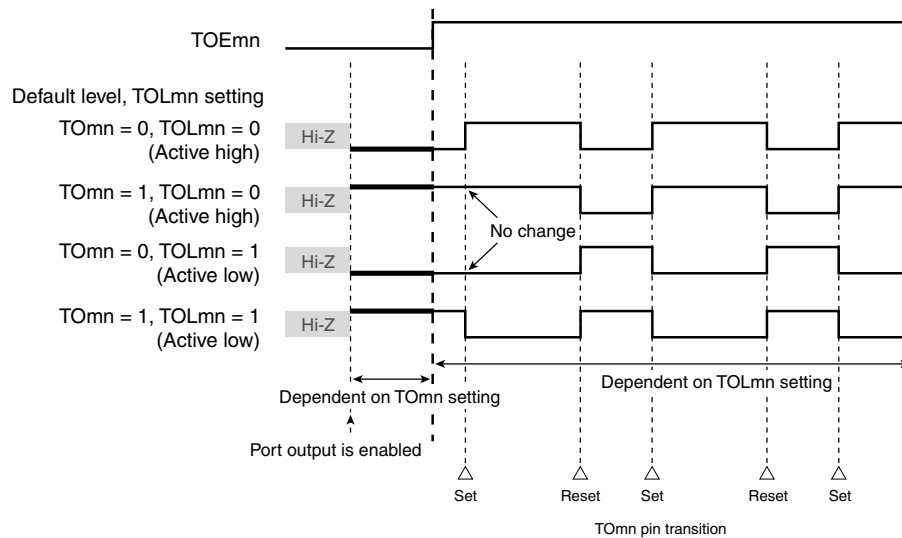
- Remarks**
1. Toggle: Reverse TOMn pin output status
  2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13



**(b) When operation starts with TOMmn = 1 setting (Combination-operation mode (PWM output))**

When TOMmn = 1, the active level is determined by TOLmn setting.

**Figure 7-29. TOMn Pin Output Status at PWM Output (TOMmn = 1)**



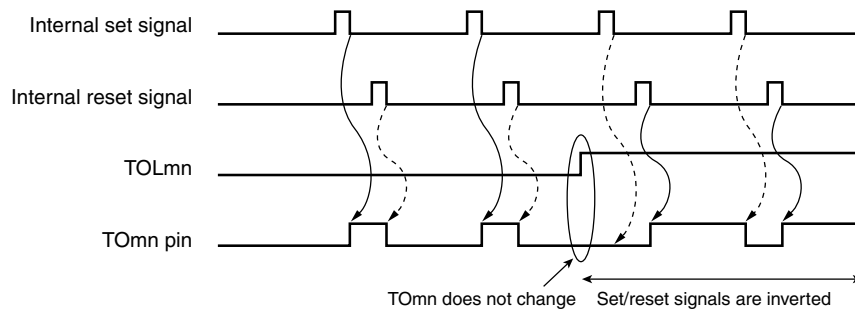
- Remarks 1.** Set: The output signal of TOMn pin changes from inactive level to active level.  
 Reset: The output signal of TOMn pin changes from active level to inactive level.
- 2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
 mn = 00 to 07, 10 to 13

**(3) Operation of TOMn pin in combination-operation mode (TOMmn = 1)**

**(a) When TOLmn setting has been changed during timer operation**

When the TOLmn setting has been changed during timer operation, the setting becomes valid at the generation timing of TOMn change condition. Rewriting TOLmn does not change the output level of TOMn. The following figure shows the operation when the value of TOLmn has been changed during timer operation (TOMmn = 1).

**Figure 7-30. Operation when TOLmn Has Been Changed during Timer Operation**



- Remarks 1.** Set: The output signal of TOMn pin changes from inactive level to active level.  
 Reset: The output signal of TOMn pin changes from active level to inactive level.
- 2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
 mn = 00 to 07, 10 to 13

**(b) Set/reset timing**

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

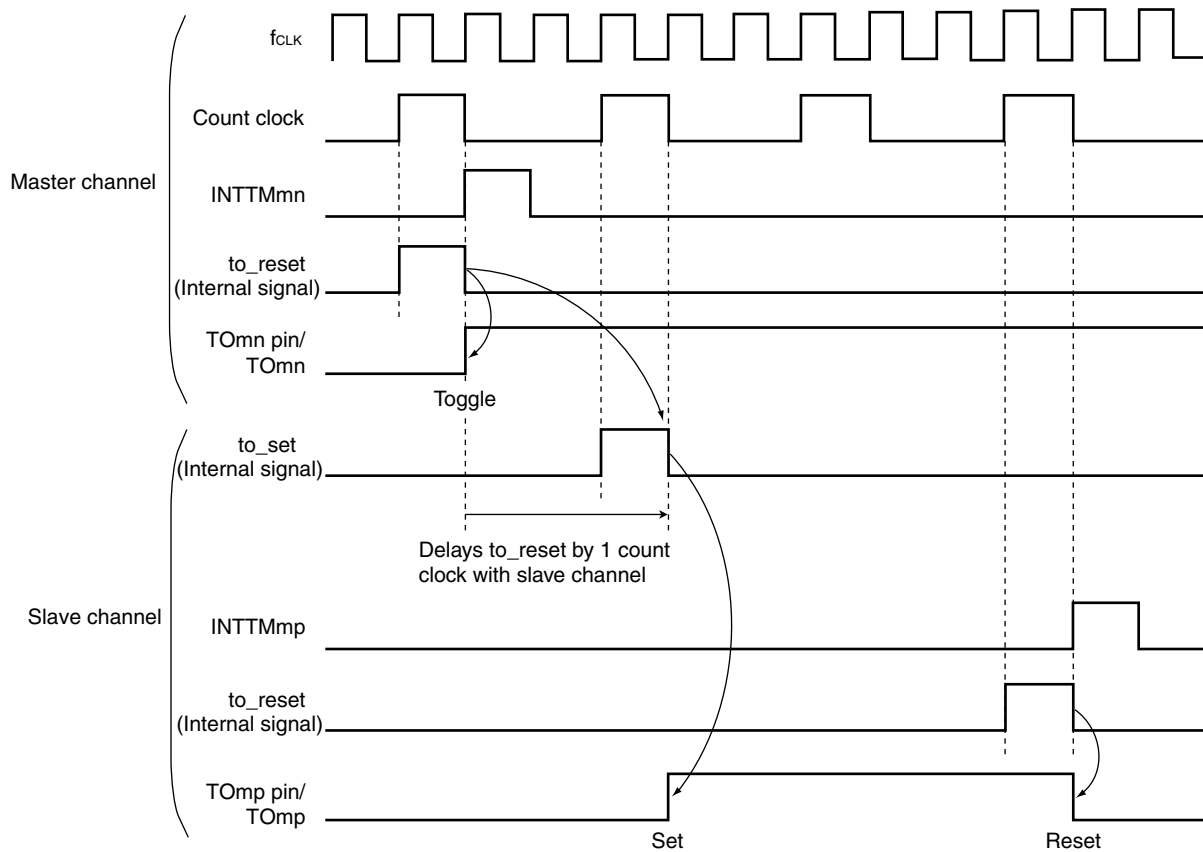
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 7-31 shows the set/reset operating statuses where the master/slave channels are set as follows.

Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0

Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

**Figure 7-31. Set/Reset Timing Operating Statuses**

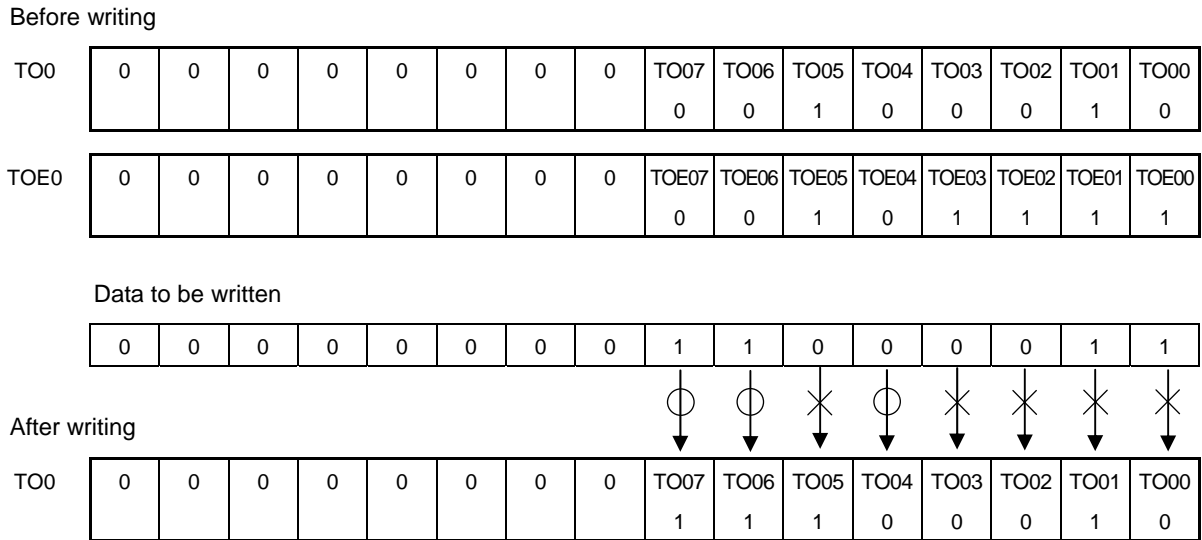


- Remarks 1.** to\_reset: TOmn pin reset/toggle signal  
to\_set: TOmn pin set signal
- 2.** m: Unit number, n: Channel number, p: Slave channel number
- When m = 0  
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)  
n < p ≤ 7 (where p is a consecutive integer greater than n)
- When m = 1  
n = 0 to 3 (n = 0, 2 for master channel)  
n < p ≤ 3 (where p is a consecutive integer greater than n)

7.4.4 Collective manipulation of TO<sub>m</sub> bits

In the TO<sub>m</sub> register, the setting bits for all the channels are located in one register in the same way as the TSm register (channel start trigger). Therefore, TO<sub>m</sub>n of all the channels can be manipulated collectively. Only specific bits can also be manipulated by setting the corresponding TOE<sub>m</sub>n = 0 to a target TO<sub>m</sub>n (channel output).

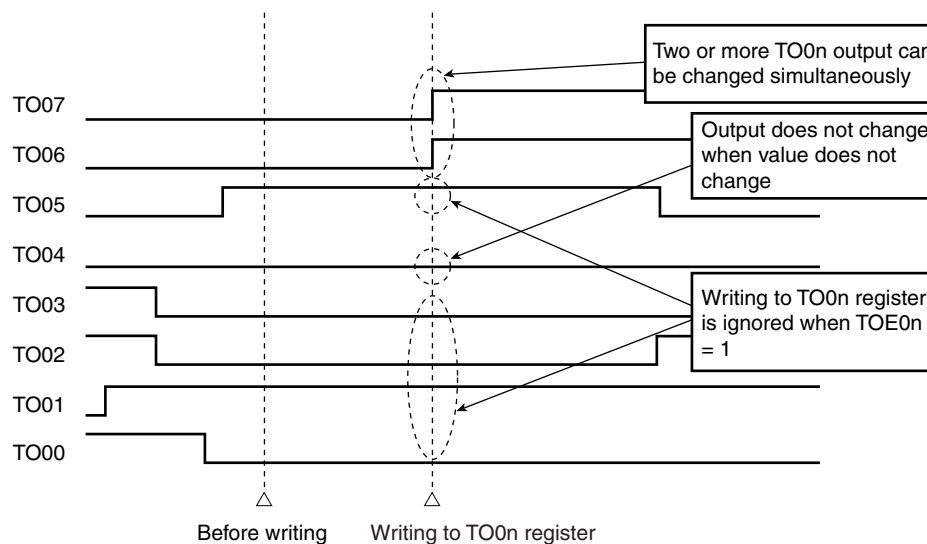
Figure 7-32. Example of TO<sub>0</sub>n Bits Collective Manipulation



Writing is done only to TO<sub>m</sub>n bits with TOE<sub>m</sub>n = 0, and writing to TO<sub>m</sub>n bits with TOE<sub>m</sub>n = 1 is ignored.

TO<sub>m</sub>n (channel output) to which TOE<sub>m</sub>n = 1 is set is not affected by the write operation. Even if the write operation is done to TO<sub>m</sub>n, it is ignored and the output change by timer operation is normally done.

Figure 7-33. TO<sub>0</sub>n Pin Statuses by Collective Manipulation of TO<sub>0</sub>n Bits



(Caution and Remark are given on the next page.)

**Caution** When  $TOEmn = 1$ , even if the output by timer interrupt of each timer ( $INTTMmn$ ) contends with writing to  $TOMn$ , output is normally done to  $TOMn$  pin.

**Remark** m: Unit number ( $m = 0, 1$ ), n: Channel number ( $n = 0$  to  $7$ ),  
mn = 00 to 07, 10 to 13

#### 7.4.5 Timer Interrupt and $TOMn$ Pin Output at Operation Start

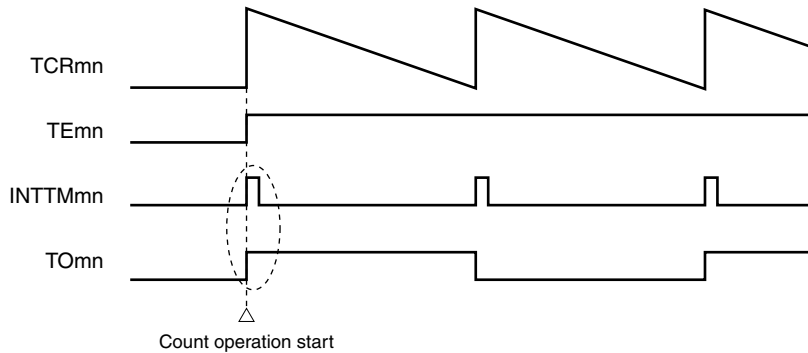
In the interval timer mode or capture mode, the  $MDmn0$  bit in the  $TMRmn$  register sets whether or not to generate a timer interrupt at count start.

When  $MDmn0$  is set to 1, the count operation start timing can be known by the timer interrupt ( $INTTMmn$ ) generation.

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

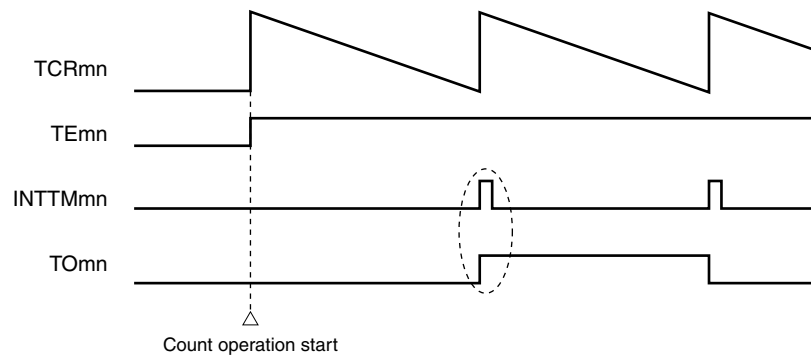
Figures 7-34 and 7-35 show operation examples when the interval timer mode ( $TOEmn = 1$ ,  $TOMmn = 0$ ) is set.

**Figure 7-34. When  $MDmn0$  is set to 1**



When  $MDmn0$  is set to 1, a timer interrupt ( $INTTMmn$ ) is output at count operation start, and  $TOMn$  performs a toggle operation.

**Figure 7-35. When  $MDmn0$  is set to 0**



When  $MDmn0$  is set to 0, a timer interrupt ( $INTTMmn$ ) is not output at count operation start, and  $TOMn$  does not change either. After counting one cycle,  $INTTMmn$  is output and  $TOMn$  performs a toggle operation.

**Remark** m: Unit number ( $m = 0, 1$ ), n: Channel number ( $n = 0$  to  $7$ ),  
mn = 00 to 07, 10 to 13

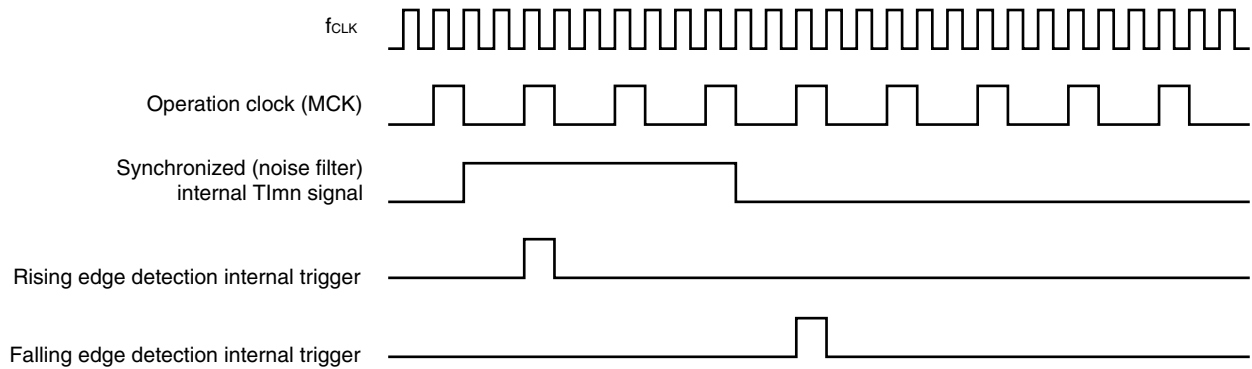
## 7.5 Channel Input (Tl<sub>mn</sub> Pin) Control

### 7.5.1 Tl<sub>mn</sub> edge detection circuit

#### (1) Edge detection basic operation timing

Edge detection circuit sampling is done in accordance with the operation clock (MCK).

**Figure 7-36. Edge Detection Basic Operation Timing**



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

## 7.6 Basic Function of Timer Array Unit

### 7.6.1 Overview of single-operation function and combination-operation function

The timer array unit consists of several channels and has a single-operation function that allows each channel to operate independently, and a combination-operation function that uses two or more channels in combination.

The single-operation function can be used for any channel, regardless of the operation mode of the other channels.

The combination-operation function is realized by combining a master channel (reference timer that mainly counts periods) and a slave channel (timer that operates in accordance with the master channel), and several rules must be observed when using this function.

### 7.6.2 Basic rules of combination-operation function

The basic rules of using the combination-operation function are as follows.

- (1) Only an even channel (channel 0, 2, 4, etc.) can be set as a master channel.
- (2) Any channel, except channel 0, can be set as a slave channel.
- (3) The slave channel must be lower than the master channel.

Example: If channel 2 of the TAU0 is set as a master channel, channel 3 or those that follow (channels 3, 4, 5, etc.) can be set as a slave channel.

If channel 2 of the TAU1 is set as a master channel, channel 3 (only up to channel 3 on TAU1.) can be set as a slave channel.

- (4) Two or more slave channels can be set for one master channel.
- (5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.  
Example: If channels 0 and 4 of the TAU0 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.

- (6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKS bit (bit 15 of the TMRmn register) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
- (7) A master channel can transmit INTTMmn (interrupt), start software trigger, and count clock to the lower channels.
- (8) A slave channel can use the INTTMmn (interrupt), start software trigger, and count clock of the master channel, but it cannot transmit its own INTTMmn (interrupt), start software trigger, and count clock to the lower channel.
- (9) A master channel cannot use the INTTMmn (interrupt), start software trigger, and count clock from the other master channel.
- (10) To simultaneously start channels that operate in combination, the TSmn bit of the channels in combination must be set at the same time.
- (11) During a counting operation, the TSmn bit of all channels that operate in combination or only the master channel can be set. TSmn of only a slave channel cannot be set.
- (12) To stop the channels in combination simultaneously, the TTmn bit of the channels in combination must be set at the same time.

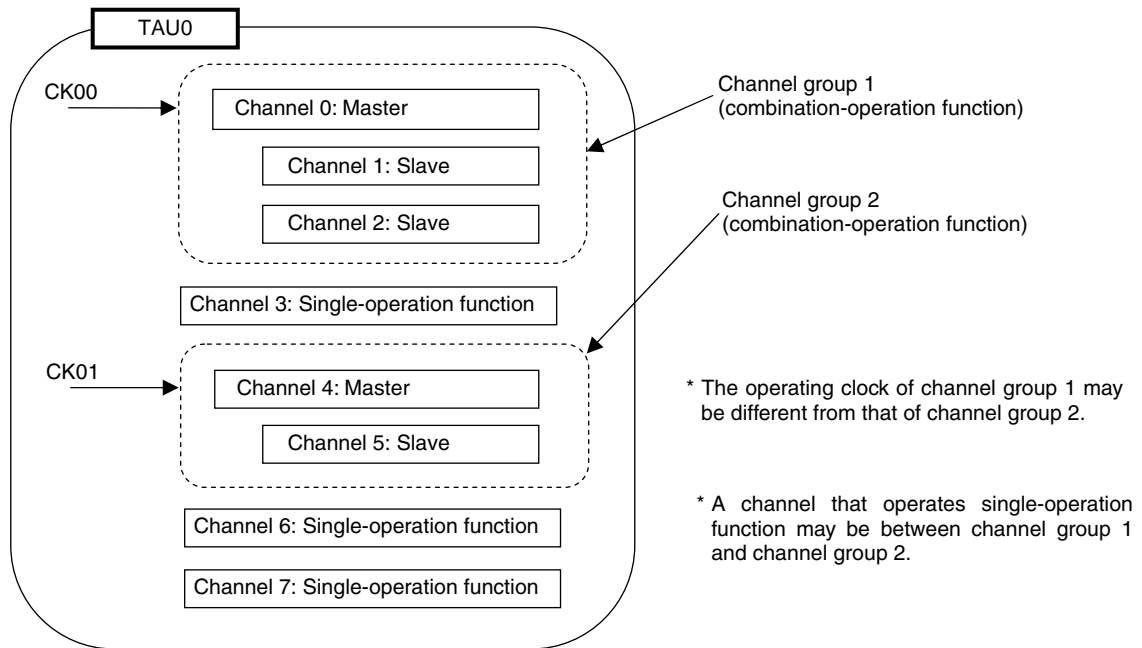
**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

### 7.6.3 Applicable range of basic rules of combination-operation function

The rules of the combination-operation function are applied in a channel group (a master channel and slave channels forming one combination-operation function).

If two or more channel groups that do not operate in combination are specified, the basic rules of the combination-operation function in 7.6.2 **Basic rules of combination-operation function** do not apply to the channel groups.

Example



## 7.7 Operation of Timer Array Unit as Independent Channel

### 7.7.1 Operation as interval timer/square wave output

#### (1) Interval timer

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

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

$$\text{Generation period of INTTMmn (timer interrupt)} = \text{Period of count clock} \times (\text{Set value of TDRmn} + 1)$$

A subsystem clock divided by four ( $f_{\text{SUB}}/4$ ) can be selected as the count clock, in addition to CKm0 and CKm1. Consequently, the interval timer can be operated with the count clock fixed to  $f_{\text{SUB}}/4$ , regardless of the  $f_{\text{CLK}}$  frequency (main system clock, subsystem clock). When changing the clock selected as  $f_{\text{CLK}}$  (changing the value of the system clock control register (CKC)), however, stop the timer array unit m (TAUm) (TT0 = 00FFH, TT1 = 000FH) first.

#### (2) Operation as square wave output

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

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

$$\bullet \text{ Period of square wave output from TOmn} = \text{Period of count clock} \times (\text{Set value of TDRmn} + 1) \times 2$$

$$\bullet \text{ Frequency of square wave output from TOmn} = \text{Frequency of count clock} / \{(\text{Set value of TDRmn} + 1) \times 2\}$$

TCRmn operates as a down counter in the interval timer mode.

TCRmn loads the value of TDRmn at the first count clock after the channel start trigger bit (TSmn) is set to 1. If MDmn0 of TMRmn = 0 at this time, INTTMmn is not output and TOmn is not toggled. If MDmn0 of TMRmn = 1, INTTMmn is output and TOmn is toggled.

After that, TCRmn count down in synchronization with the count clock.

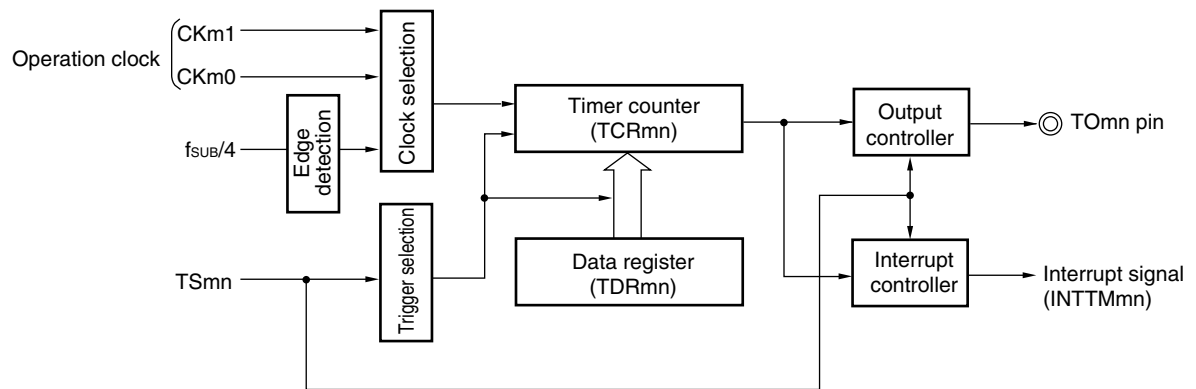
When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, TCRmn loads the value of TDRmn again. After that, the same operation is repeated.

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

- Remarks**
1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13
  2.  $f_{\text{CLK}}$ : CPU/peripheral hardware clock frequency  
 $f_{\text{SUB}}$ : Subsystem clock oscillation frequency

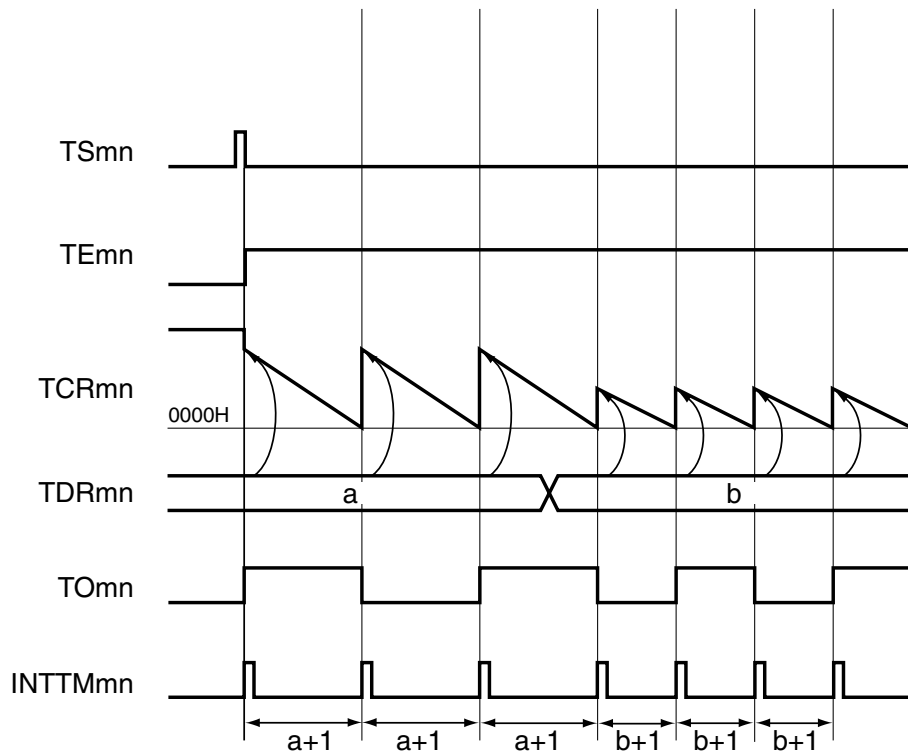


Figure 7-37. Block Diagram of Operation as Interval Timer/Square Wave Output



- Remarks 1.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13
- 2.** f<sub>SUB</sub>: Subsystem clock oscillation frequency

Figure 7-38. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)

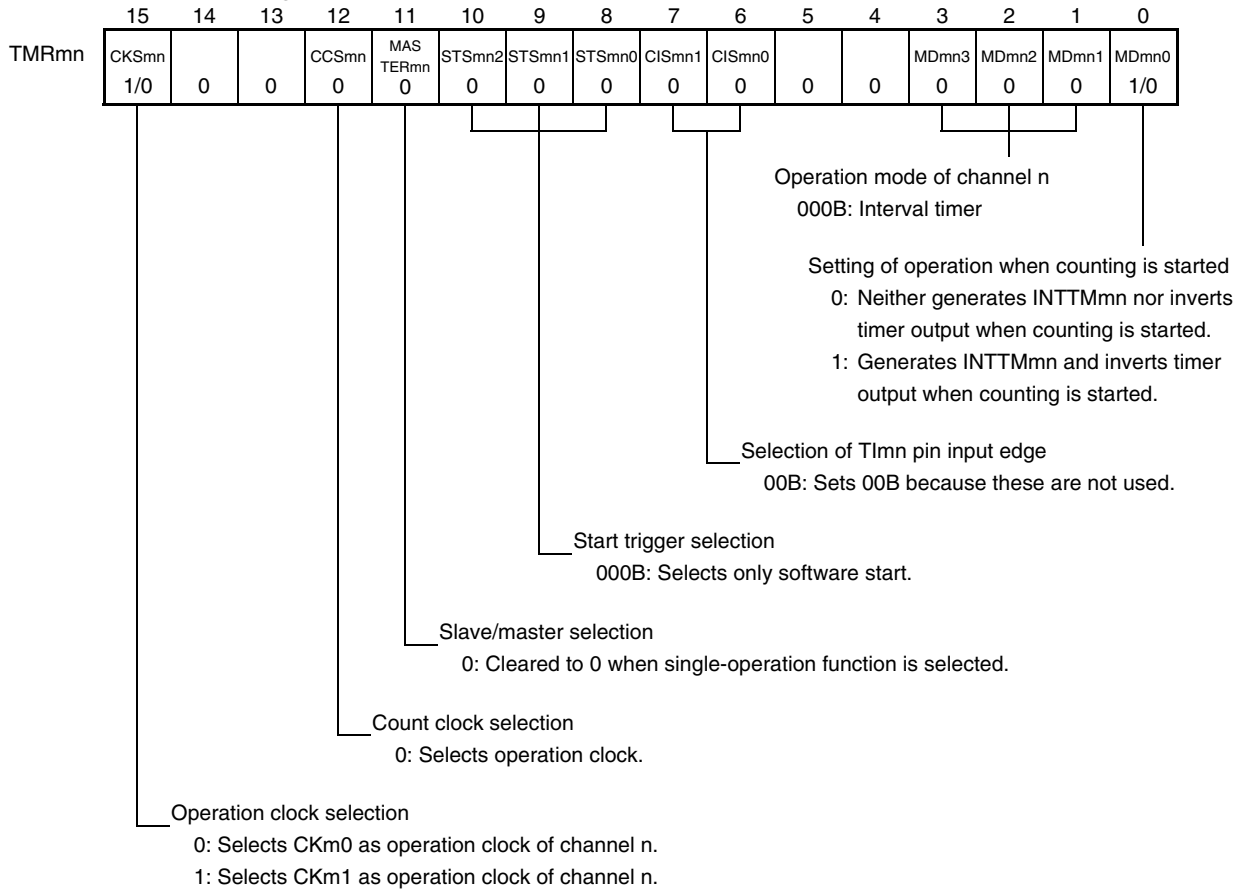


- Remarks**
1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13
  2.  $f_{SUB}$ : Subsystem clock oscillation frequency

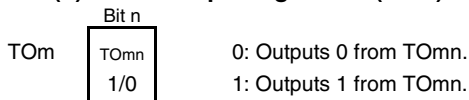
Figure 7-39. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output(1/3)

(1) When CKm0 or CKm1 is selected as count clock

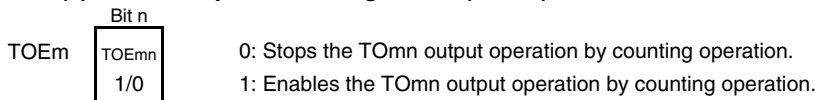
(a) Timer mode register mn (TMRmn)



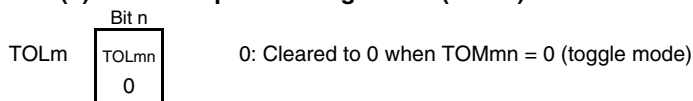
(b) Timer output register m (TOM)



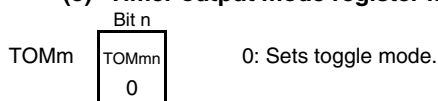
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)

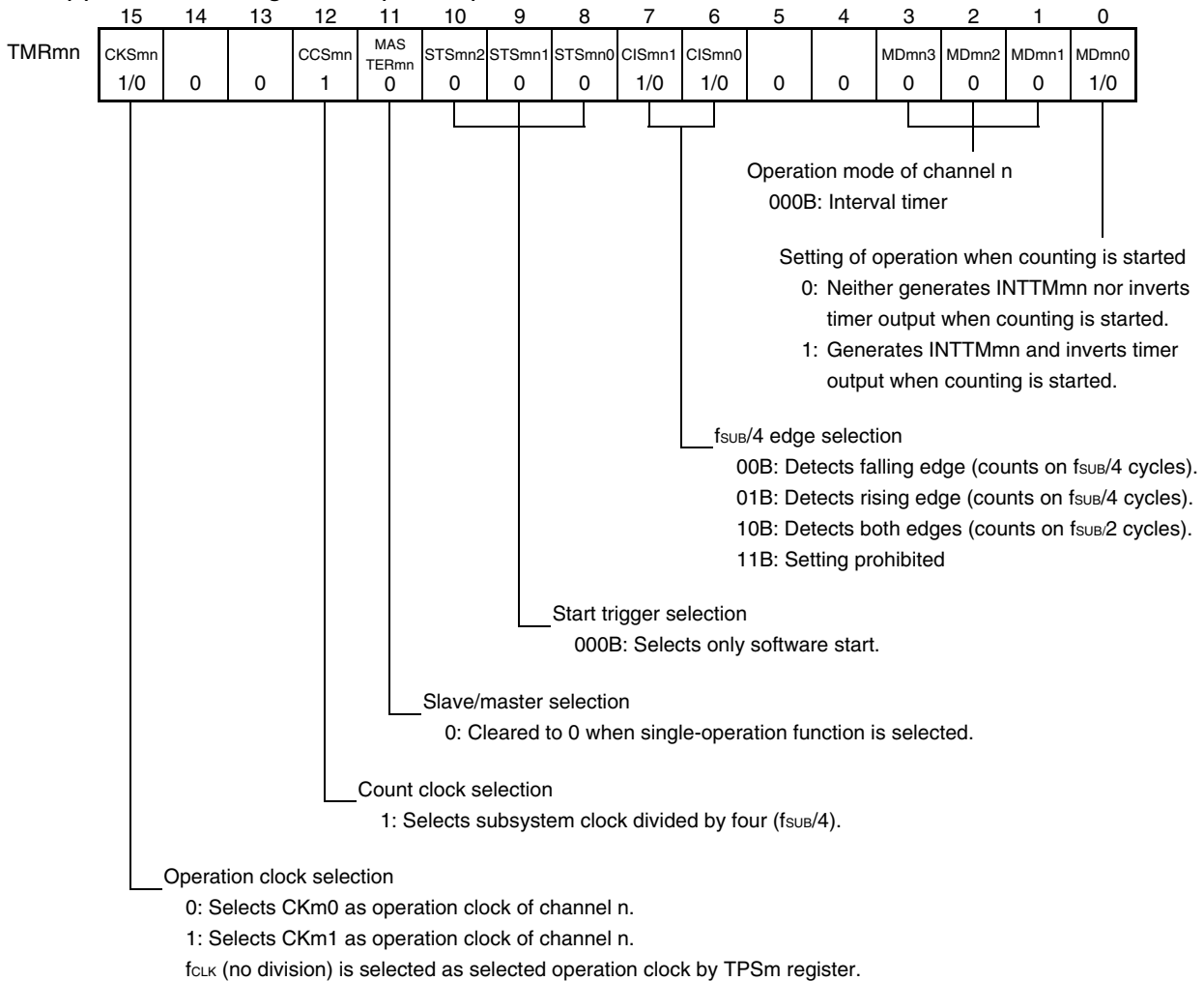


**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 13

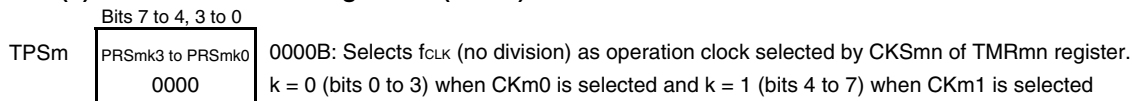
Figure 7-39. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/3)

(2) When  $f_{SUB}/4$  is selected as count clock

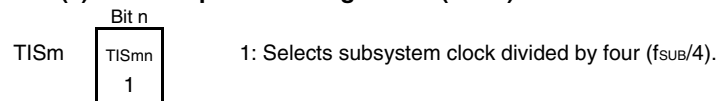
(a) Timer mode register mn (TMRmn)



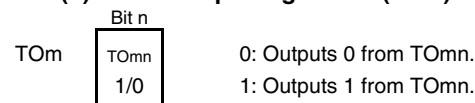
(b) Timer clock select register m (TPSm)



(c) Timer input select register m (TISm)



(d) Timer output register m (TOM)



- Remarks**
1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 13, k = 0, 1
  2.  $f_{SUB}$ : Subsystem clock oscillation frequency

**Figure 7-39. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (3/3)**

**(2) When  $f_{SUB}/4$  is selected as count clock (continued)**

**(e) Timer output enable register m (TOEm)**

	Bit n	
TOEm	TOEmn 1/0	<p>0: Stops the TOMn output operation by counting operation.</p> <p>1: Enables the TOMn output operation by counting operation.</p>

**(f) Timer output level register m (TOLm)**

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (toggle mode)

**(g) Timer output mode register m (TOMm)**

	Bit n	
TOMm	TOMmn 0	0: Sets toggle mode.

- Remarks**
1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13
  2.  $f_{SUB}$ : Subsystem clock oscillation frequency

Figure 7-40. 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, TAU1EN bit of the PER0 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel). Sets the TISmn bit to 1 ( $f_{SUB}/4$ ) when $f_{SUB}/4$ is selected as the count clock. Sets interval (period) value to the TDRmn register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOMn output Clears the TOMmn bit of the TOMm register to 0 (toggle mode). Clears the TOLmn bit to 0. Sets the TOMn bit and determines default level of the TOMn output.	The TOMn pin goes into Hi-Z output state.  The TOMn default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets TOEmn to 1 and enables operation of TOMn. Clears the port register and port mode register to 0.	TOMn does not change because channel stops operating. The TOMn pin outputs the TOMn set level.
Operation start	Sets TOEmn to 1 (only when operation is resumed). Sets the TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of TDRmn is loaded to TCRmn at the count clock input. INTTMmn is generated and TOMn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set values of the TMRmn register, TOMmn, and TOLmn bits cannot be changed. Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOM and TOEm registers can be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of TDRmn is loaded to TCRmn again and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOMn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn holds count value and stops. The TOMn output is not initialized but holds current status.
	TOEmn is cleared to 0 and value is set to TOMn bit.	The TOMn pin outputs the TOMn set level.

Operation is resumed.

- Remark**
- m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13
  - $f_{SUB}$ : Subsystem clock oscillation frequency

**Figure 7-40. Operation Procedure of Interval Timer/Square Wave Output Function (2/2)**

	Software Operation	Hardware Status
TAU stop	To hold the TOMn pin output level Clears TOMn bit to 0 after the value to be held is set to the port register.	The TOMn pin output level is held by port function.
	When holding the TOMn pin output level is not necessary Switches the port mode register to input mode.	The TOMn pin output level goes into Hi-Z output state.
	The TAU0EN bit, TAU1EN bit of the PER0 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOMn bit is cleared to 0 and the TOMn pin is set to port mode.)

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

7.7.2 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 TImn pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

$$\text{Specified number of counts} = \text{Set value of TDRmn} + 1$$

TCRmn operates as a down counter in the event counter mode.

When the channel start trigger bit (TSmn) is set to 1, TCRmn loads the value of TDRmn.

TCRmn counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, TCRmn loads the value of TDRmn again, and outputs INTTMmn.

After that, the above operation is repeated.

TOmn must not be used because its waveform depends on the external event and irregular.

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

Figure 7-41. Block Diagram of Operation as External Event Counter

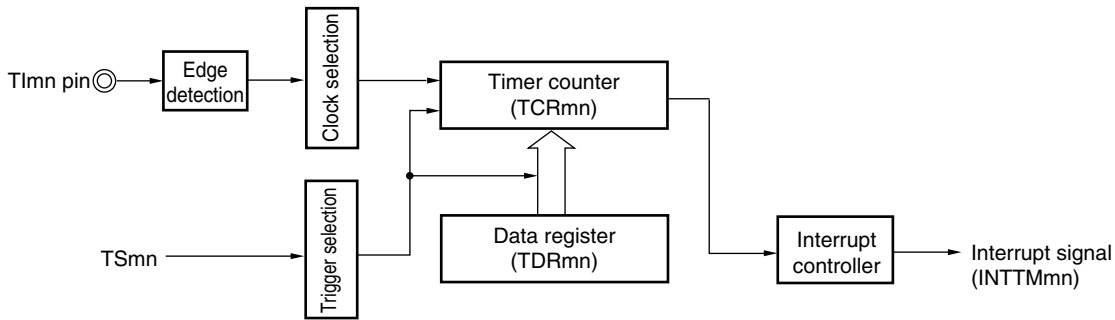
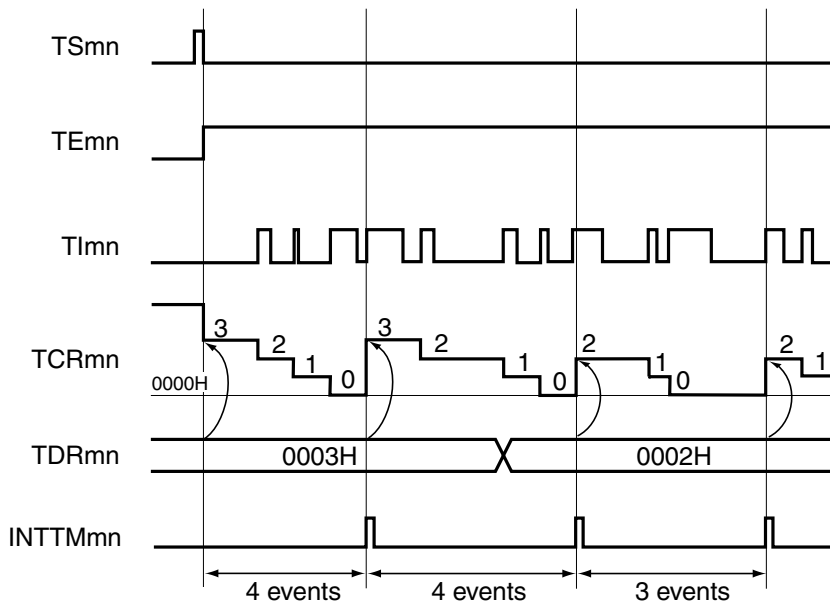


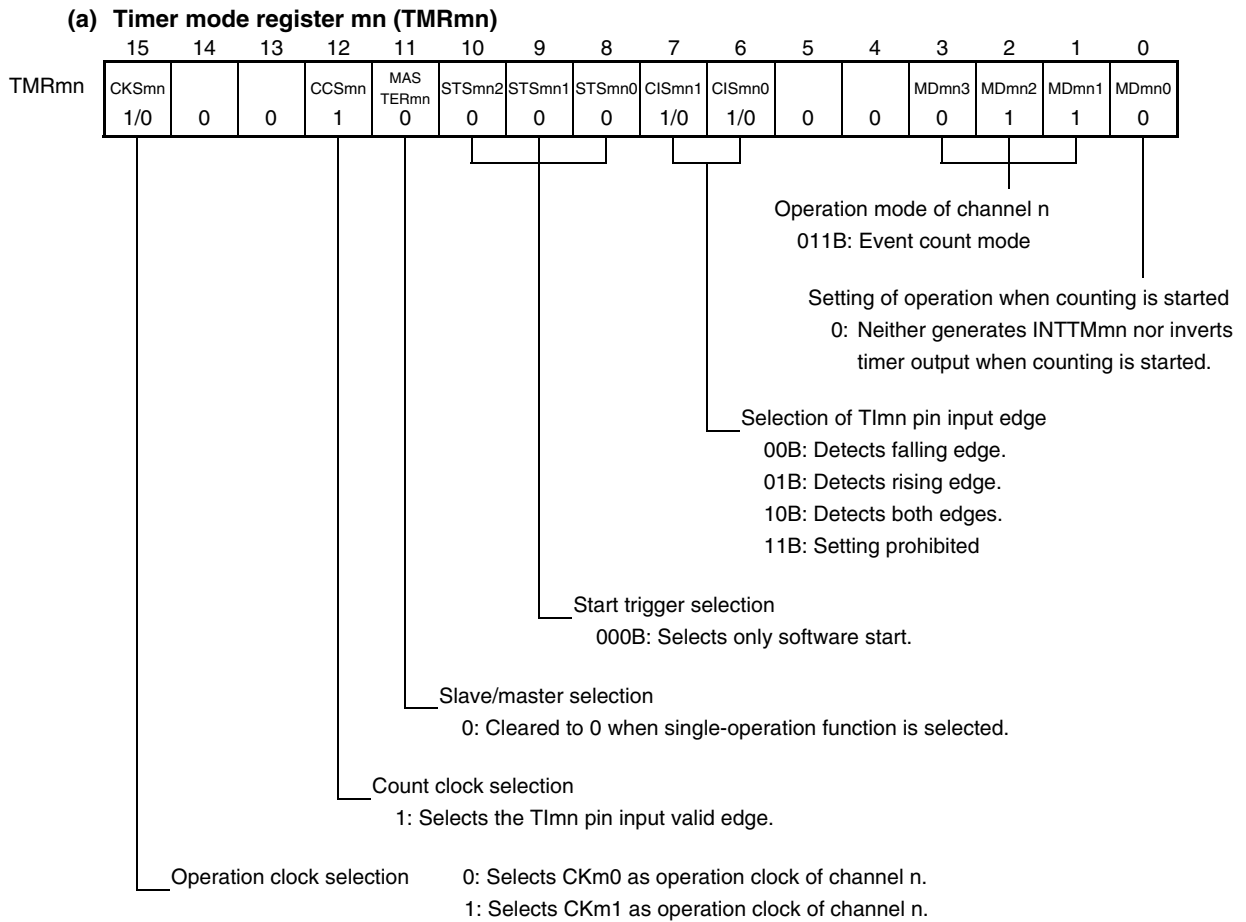
Figure 7-42. Example of Basic Timing of Operation as External Event Counter



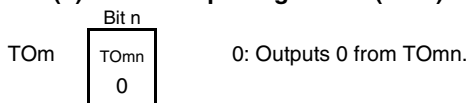
**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 13



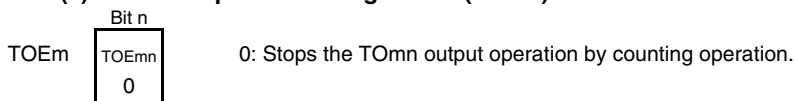
Figure 7-43. Example of Set Contents of Registers in External Event Counter Mode



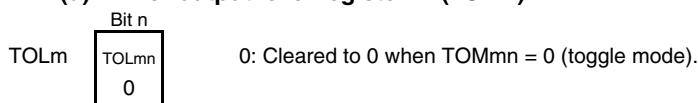
(b) **Timer output register m (TOM)**



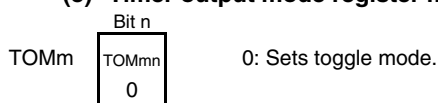
(c) **Timer output enable register m (TOEm)**



(d) **Timer output level register m (TOLm)**



(e) **Timer output mode register m (TOMm)**



**Remark**    m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
 mn = 00 to 07, 10 to 13

Figure 7-44. 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, TAU1EN bit of the PER0 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel). Sets number of counts to the TDRmn register. Clears the TOEmn bit of the TOEm register to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of TDRmn is loaded to TCRmn and detection of the TImn pin input edge is awaited.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts down each time input edge of the TImn pin has been detected. When count value reaches 0000H, the value of TDRmn is loaded to TCRmn again, and the count operation is continued. By detecting TCRmn = 0000H, the INTTMmn output is generated. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn holds count value and stops.
TAU stop	The TAU0EN bit, TAU1EN bit of the PER0 register are cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

### 7.7.3 Operation as frequency divider (channel 0 of unit 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 TO00.

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)

TCR00 operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS00) is set to 1, TCR00 loads the value of TDR00 when the TI00 valid edge is detected. If MD000 of TMR00 = 0 at this time, INTTM00 is not output and TO00 is not toggled. If MD000 of TMR00 = 1, INTTM00 is output and TO00 is toggled.

After that, TCR00 counts down at the valid edge of TI00. When TCR00 = 0000H, it toggles TO00. At the same time, TCR00 loads the value of TDR00 again, and continues counting.

If detection of both the edges of TI00 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.

$$\text{Clock period of TO00 output} = \text{Ideal TO00 output clock period} \pm \text{Operation clock period (error)}$$

TDR00 can be rewritten at any time. The new value of TDR00 becomes valid during the next count period.

**Figure 7-45. Block Diagram of Operation as Frequency Divider**

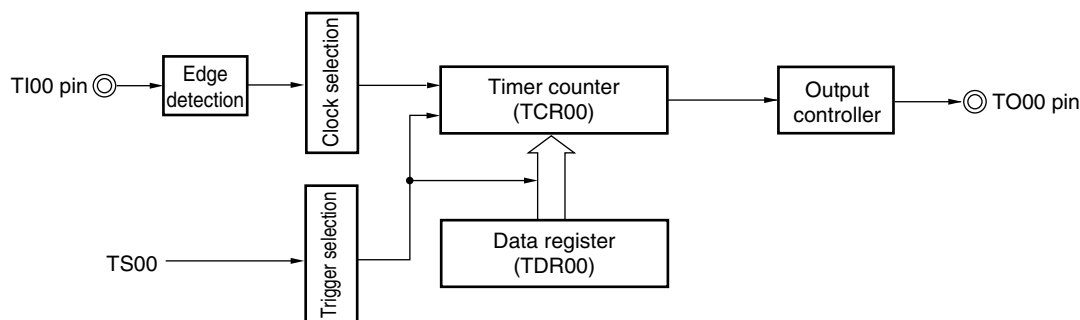


Figure 7-46. Example of Basic Timing of Operation as Frequency Divider (MD000 = 1)

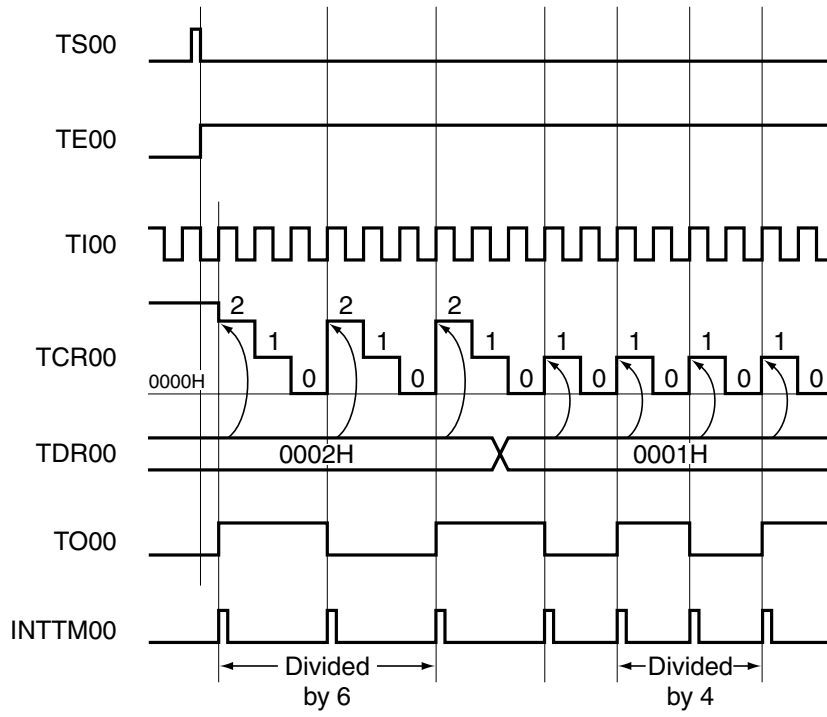
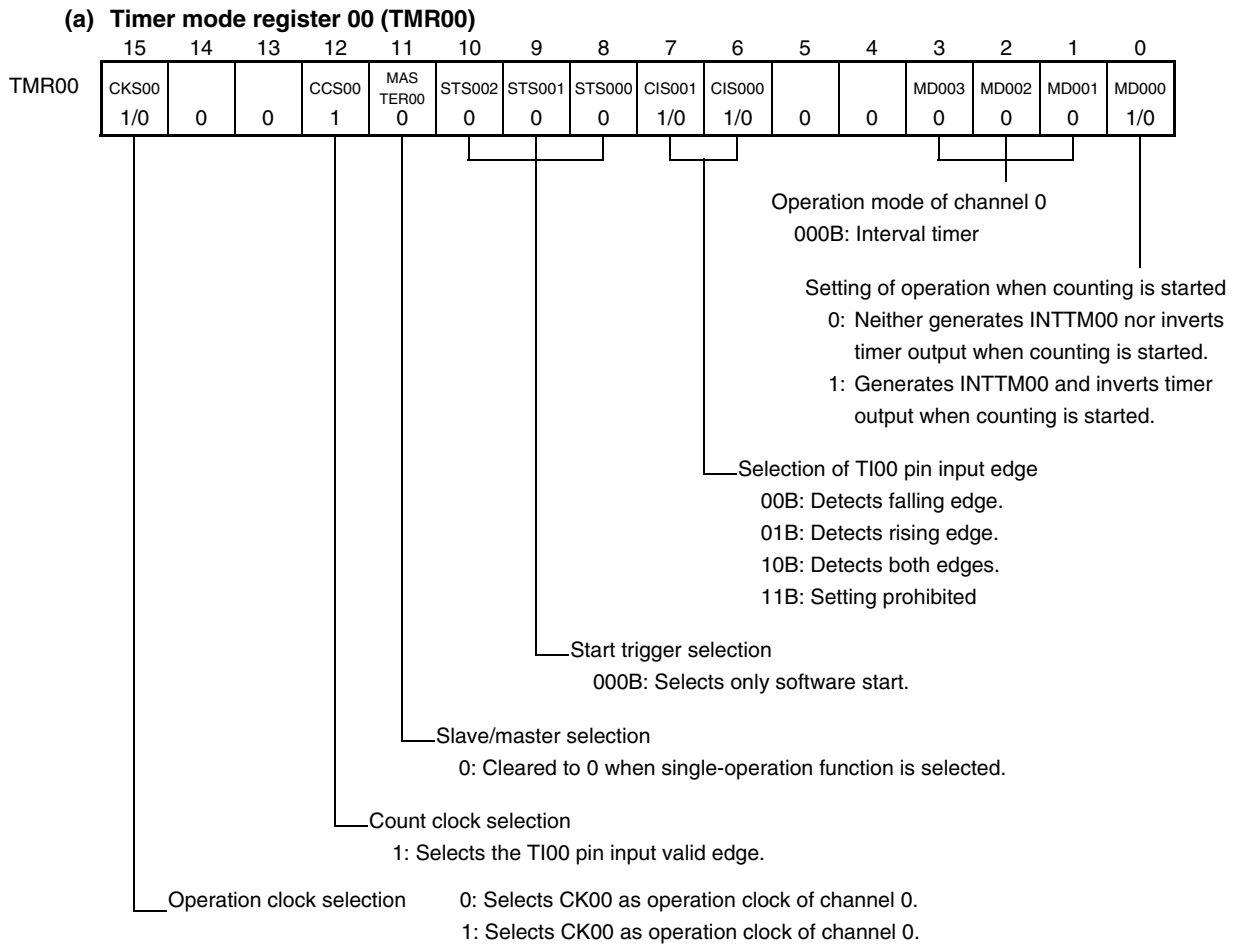
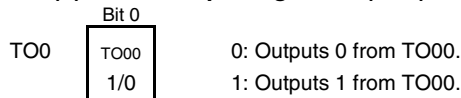


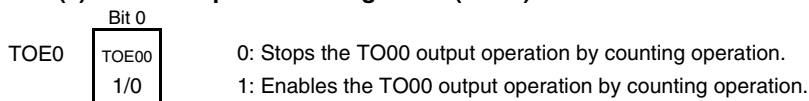
Figure 7-47. Example of Set Contents of Registers When Frequency Divider Is Used



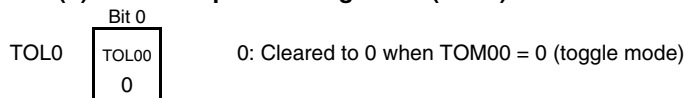
(b) **Timer output register 0 (TO0)**



(c) **Timer output enable register 0 (TOE0)**



(d) **Timer output level register 0 (TOL0)**



(e) **Timer output mode register 0 (TOM0)**

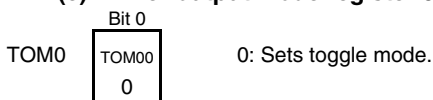


Figure 7-48. Operation Procedure When Frequency Divider Function Is Used

	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 the PER0 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR00 register (determines operation mode of channel). Sets interval (period) value to the TDR00 register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOM00 bit of the TOM0 register to 0 (toggle mode). Clears the TOL00 bit to 0. Sets the TO00 bit and determines default level of the TO00 output.	The TO00 pin goes into Hi-Z output state.  The TO00 default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOE00 to 1 and enables operation of TO00. Clears the port register and port mode register to 0.	TO00 does not change because channel stops operating. The TO00 pin outputs the TO00 set level.
Operation start	Sets the TOE00 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 TDR00 is loaded to 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 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 TDR00 is loaded to TCR00 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. TCR00 holds count value and stops. The TO00 output is not initialized but holds current status.
	TOE00 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 TO00 bit to 0 after the value to be held is set to the port register.	The TO00 pin output level is held by port function.
	When holding the TO00 pin output level is not necessary Switches the port mode register to input mode.	The TO00 pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER0 register are 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.

### 7.7.4 Operation as input pulse interval measurement

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

The pulse interval can be calculated by the following expression.

$$\text{TImn input pulse interval} = \text{Period of count clock} \times ((10000\text{H} \times \text{TSRmn: OVF}) + (\text{Capture value of TDRmn} + 1))$$

**Caution** The TImn pin input is sampled using the operating clock selected with the CKSmn bit of the TMRmn register, so an error equal to the number of operating clocks occurs.

TCRmn operates as an up counter in the capture mode.

When the channel start trigger (TSmn) is set to 1, TCRmn counts up from 0000H in synchronization with the count clock.

When the TImn pin input valid edge is detected, the count value is transferred (captured) to TDRmn and, at the same time, the counter (TCRmn) is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of the TSRmn register is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

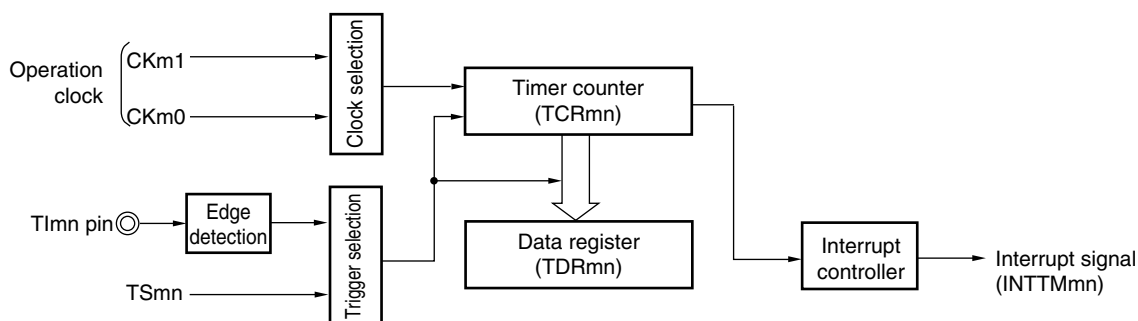
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, the OVF bit is configured as a cumulative flag, the correct interval value cannot be measured if an overflow occurs more than once.

Set STSmn2 to STSmn0 of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

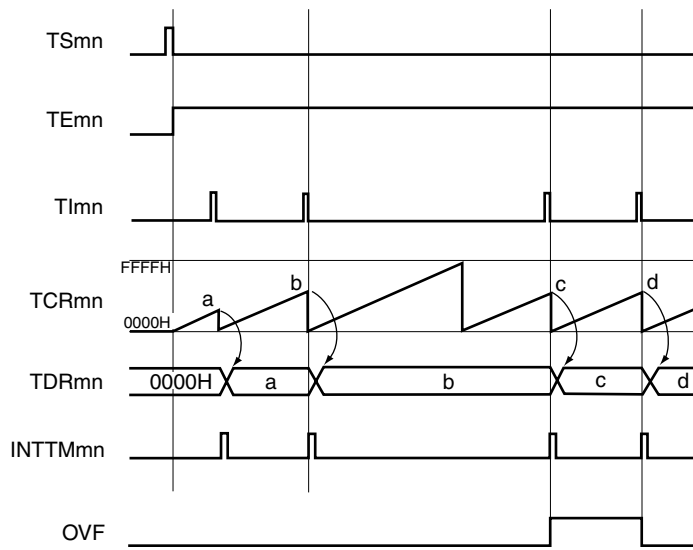
When TEMn = 1, instead of the TImn pin input, a software operation (TSmn = 1) can be used as a capture trigger.

**Figure 7-49. Block Diagram of Operation as Input Pulse Interval Measurement**



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

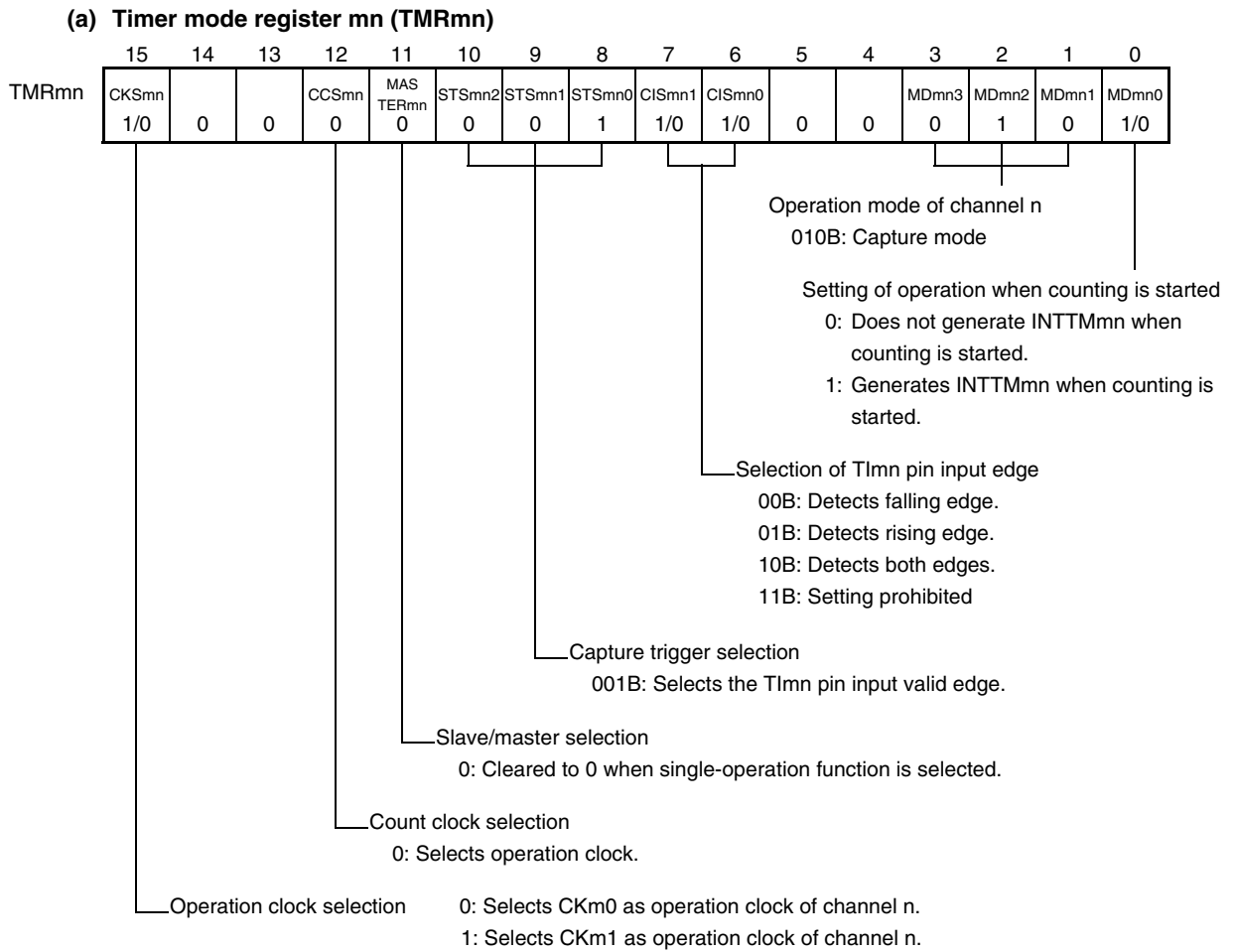
Figure 7-50. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)



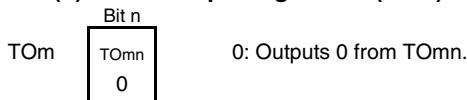
**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13



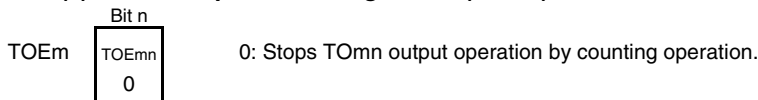
Figure 7-51. Example of Set Contents of Registers to Measure Input Pulse Interval



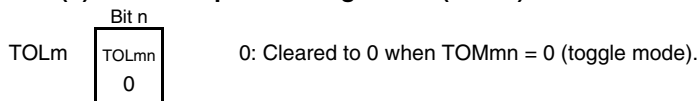
(b) Timer output register m (TOM)



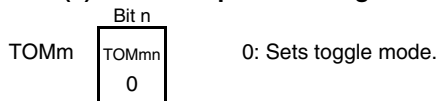
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)



(e) Timer output mode register m (TOMm)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 13

Figure 7-52. 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, TAU1EN bit of the PER0 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. TCRmn is cleared to 0000H at the count clock input. When the MDmn0 bit of the TMRmn register is 1, INTTMmn is generated.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TOMmn, TOLmn, TOMn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts up from 0000H. When the TImn pin input valid edge is detected, the count value is transferred (captured) to TDRmn. At the same time, TCRmn is cleared to 0000H, and the INTTMmn signal is generated. If an overflow occurs at this time, the OVF bit of the TSRmn register is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAU0EN bit, TAU1EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

### 7.7.5 Operation as input signal high-/low-level width measurement

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

$$\text{Signal width of TImn input} = \text{Period of count clock} \times ((10000H \times \text{TSRn: OVF}) + (\text{Capture value of TDRmn} + 1))$$

**Caution** The TImn pin input is sampled using the operating clock selected with the CKSmn bit of the TMRmn register, so an error equal to the number of operating clocks occurs.

TCRmn operates as an up counter in the capture & one-count mode.

When the channel start trigger (TSmn) is set to 1, TEMn is set to 1 and the TImn pin start edge detection wait status is set.

When the TImn start valid edge (rising edge of TImn when the high-level width is to be measured) is detected, the counter counts up in synchronization with the count clock. When the valid capture edge (falling edge of TImn when the high-level width is to be measured) is detected later, the count value is transferred to TDRmn and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of the TSRmn register is set to 1. If the counter does not overflow, the OVF bit is cleared. TCRmn stops at the value “value transferred to TDRmn + 1”, and the TImn pin start edge detection wait status is set. After that, the above operation is repeated.

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

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, the OVF bit is configured as an integral flag, and the correct interval value cannot be measured if an overflow occurs more than once.

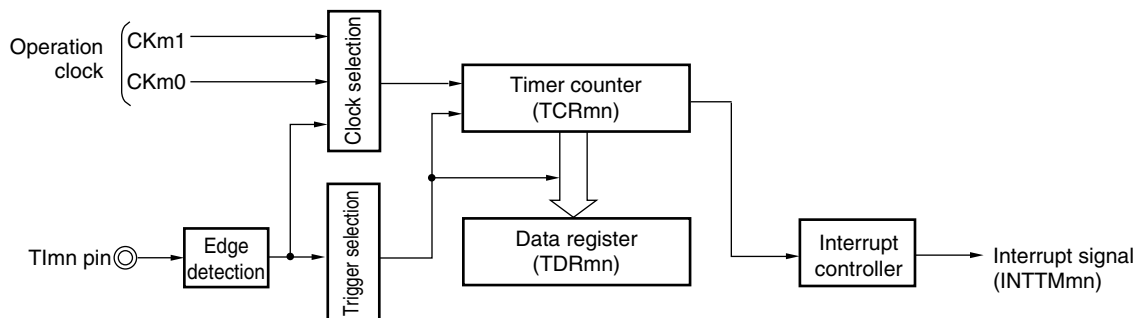
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.

Because this function is used to measure the signal width of the TImn pin input, TSmn cannot be set to 1 while TEMn is 1.

CISmn1, CISmn0 of TMRmn = 10B: Low-level width is measured.

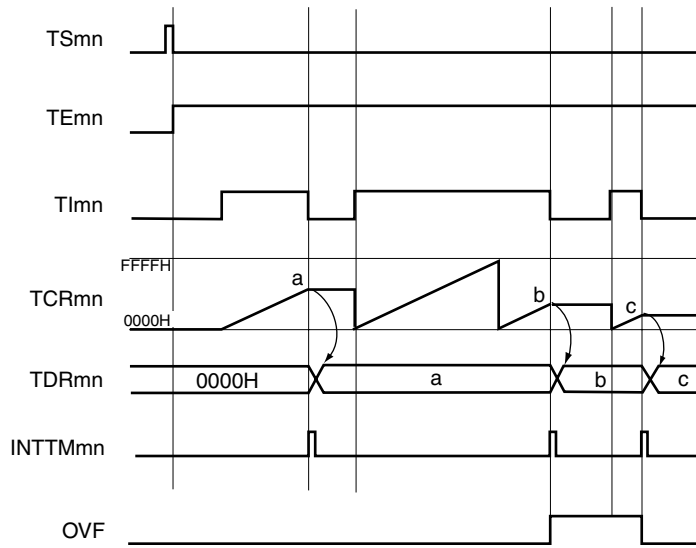
CISmn1, CISmn0 of TMRmn = 11B: High-level width is measured.

**Figure 7-53. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement**



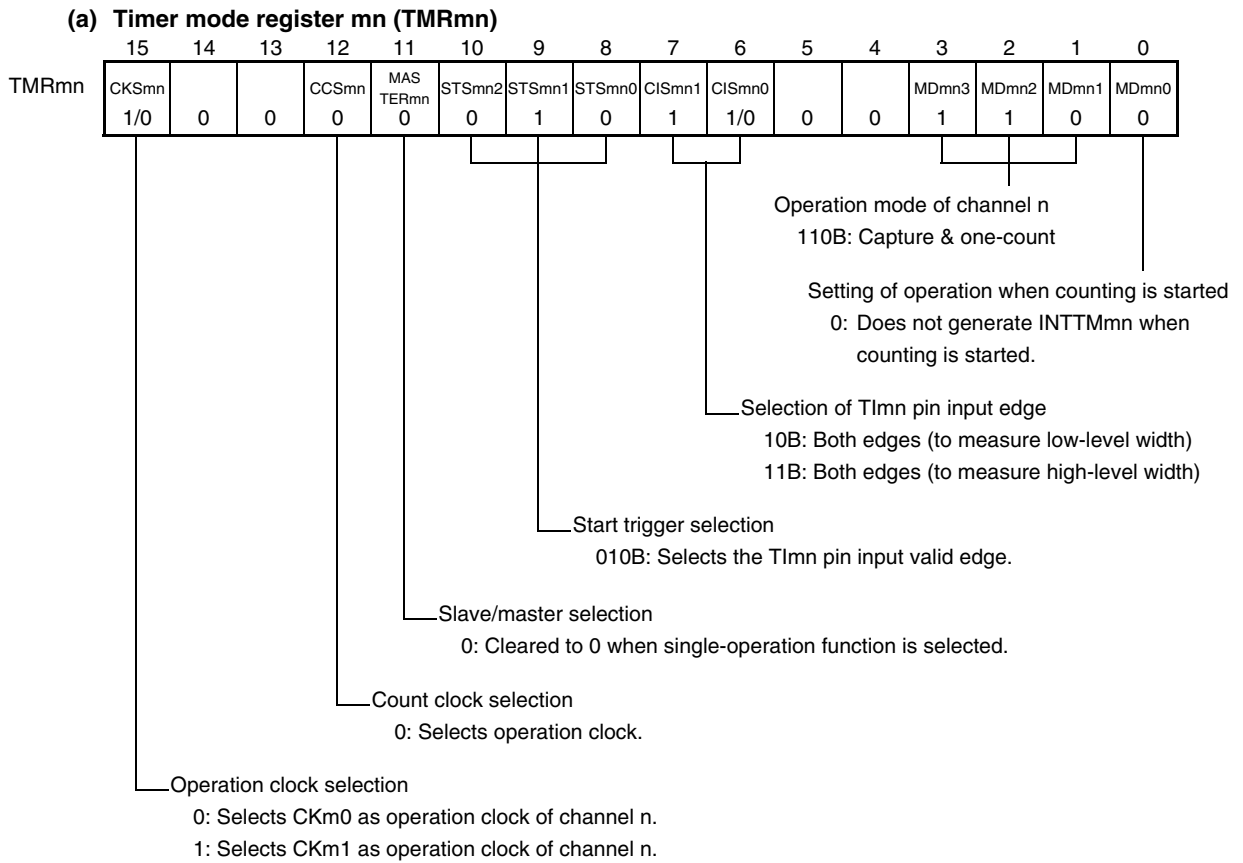
**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

Figure 7-54. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement

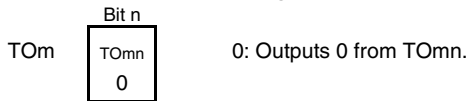


**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

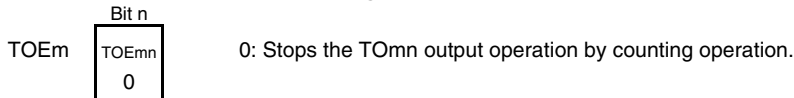
Figure 7-55. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width



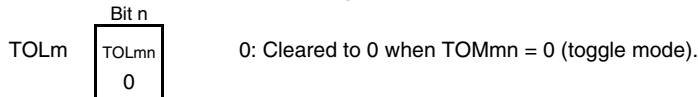
(b) **Timer output register m (TOM)**



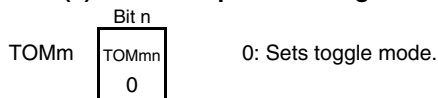
(c) **Timer output enable register m (TOEm)**



(d) **Timer output level register m (TOLm)**



(e) **Timer output mode register m (TOMm)**



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 13

Figure 7-56. 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, TAU1EN bit of the PER0 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel). Clears TOEmn to 0 and stops operation of TOMn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the TImn pin start edge detection wait status is set.
	Detects TImn pin input count start valid edge.	Clears TCRmn to 0000H and starts counting up.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOMn, and TOEmn bits cannot be changed.	When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to TDRmn and INTTMmn is generated. If an overflow occurs at this time, the OVF bit of the TSRmn register is set; if an overflow does not occur, the OVF bit is cleared. TCRmn stops the count operation until the next TImn pin start edge is detected.
Operation stop	The TTmn bit is set to 1. TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAU0EN bit, TAU1EN bit of PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),  
mn = 00 to 07, 10 to 13

## 7.8 Operation of Plural Channels of Timer Array Unit

### 7.8.1 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.

$$\begin{aligned} \text{Pulse period} &= \{\text{Set value of TDRmn (master)} + 1\} \times \text{Count clock period} \\ \text{Duty factor [\%]} &= \{\text{Set value of TDRmp (slave)}\} / \{\text{Set value of TDRmn (master)} + 1\} \times 100 \\ \text{0\% output:} & \quad \text{Set value of TDRmp (slave)} = 0000\text{H} \\ \text{100\% output:} & \quad \text{Set value of TDRmp (slave)} \geq \{\text{Set value of TDRmn (master)} + 1\} \end{aligned}$$

**Remark** The duty factor exceeds 100% if the set value of TDRmn (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode and counts the periods. When the channel start trigger (TSmn) is set to 1, INTTMmn is output. TCRmn counts down starting from the loaded value of TDRmn, in synchronization with the count clock. When TCRmn = 0000H, INTTMmn is output. TCRmn loads the value of TDRmn again. After that, it continues the similar operation.

TCRmp of a slave channel operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. TCRmp of the slave channel loads the value of TDRmp, using INTTMmn of the master channel as a start trigger, and stops counting until the next start trigger (INTTMmn of the master channel) is input.

The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

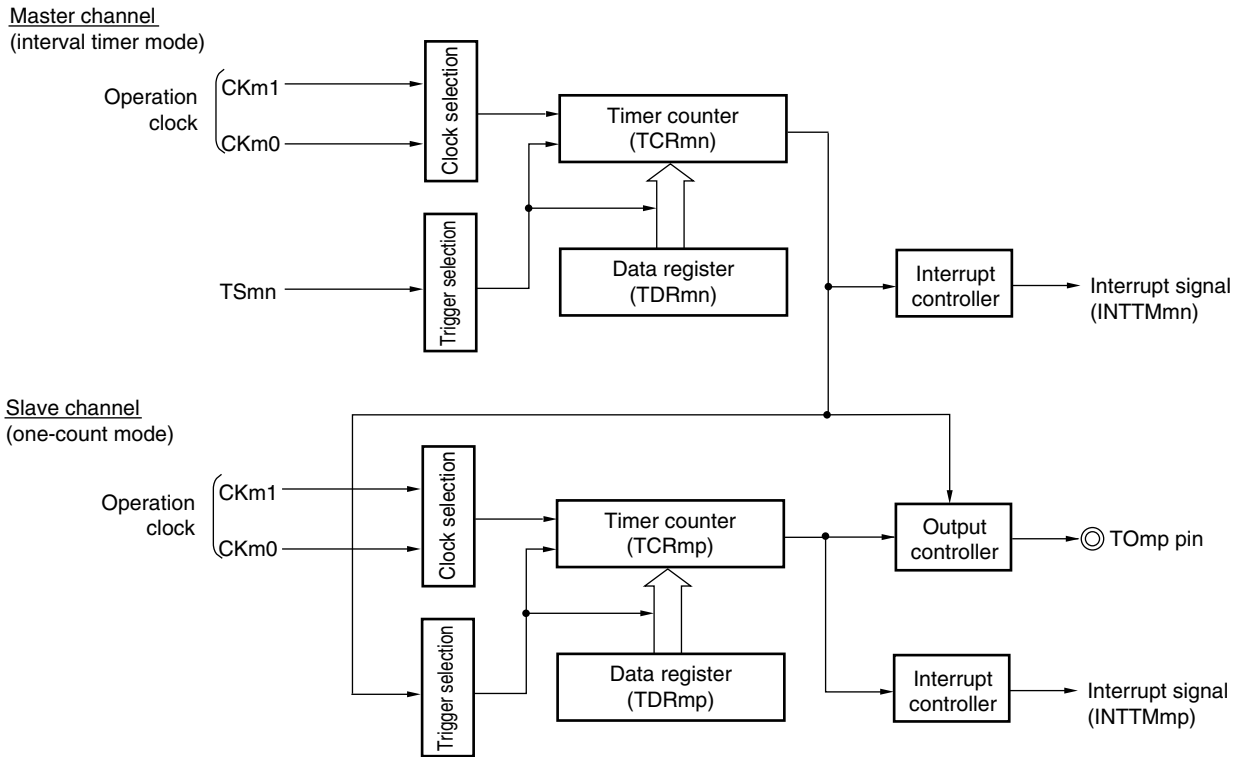
**Caution** To rewrite both TDRmn of the master channel and TDRmp of the slave channel, a write access is necessary two times. The timing at which the values of TDRmn and TDRmp are loaded to TCRmn and TRCmp is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both TDRmn of the master and TDRmp of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.

**Remark** m: Unit number, n: Channel number, p: Slave channel number (p = n+1)

When m = 0: n = 0, 2, 4, 6

When m = 1: n = 0, 2

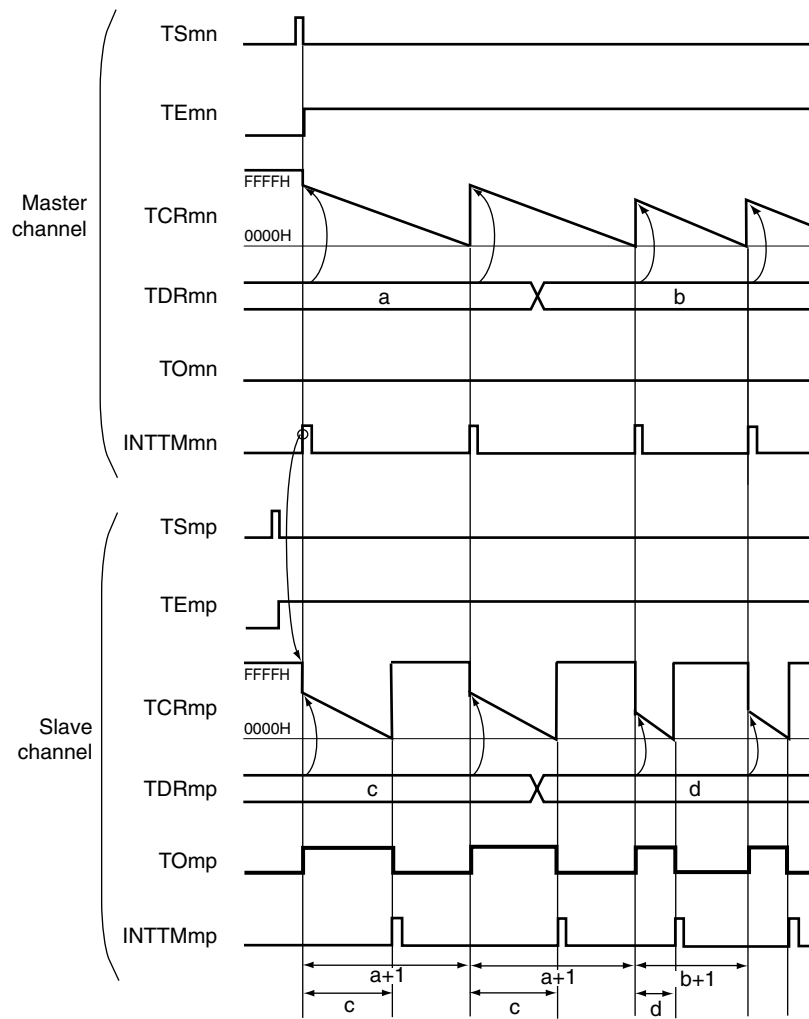
Figure 7-57. Block Diagram of Operation as PWM Function



**Remark** m: Unit number, n: Channel number, p: Slave channel number (p = n+1)  
 When m = 0: n = 0, 2, 4, 6  
 When m = 1: n = 0, 2

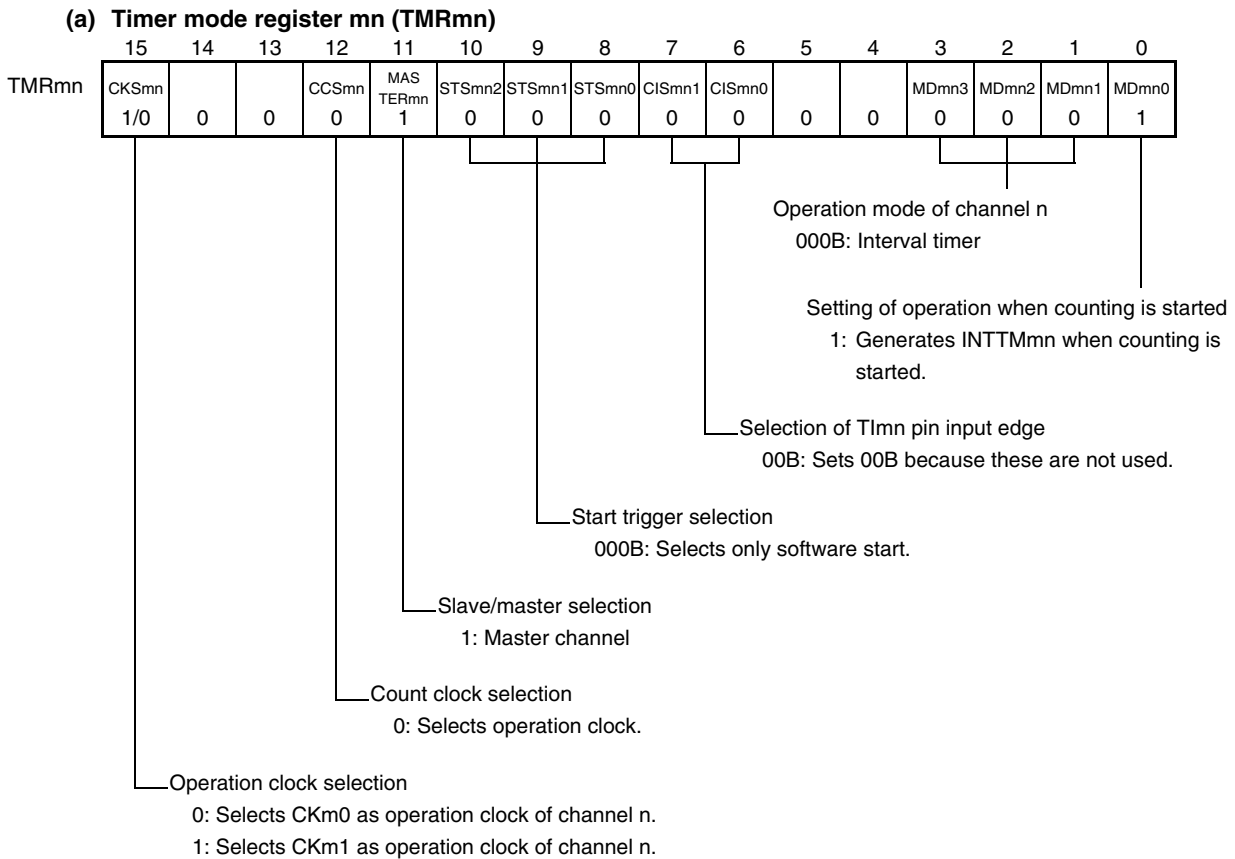


Figure 7-58. Example of Basic Timing of Operation as PWM Function

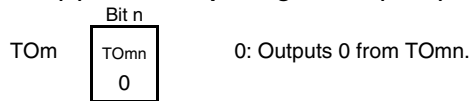


**Remark** m: Unit number, n: Channel number, p: Slave channel number ( $p = n+1$ )  
 When  $m = 0$ :  $n = 0, 2, 4, 6$   
 When  $m = 1$ :  $n = 0, 2$

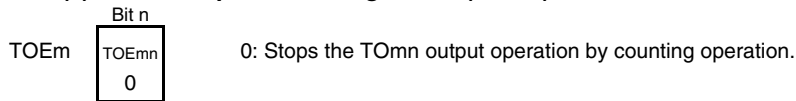
Figure 7-59. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used



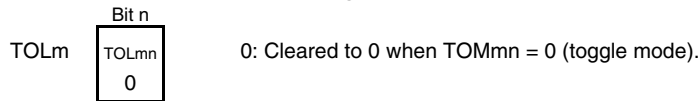
(b) Timer output register m (TOM)



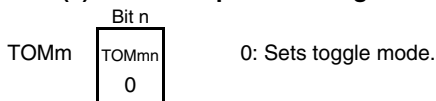
(c) Timer output enable register m (TOEm)



(d) Timer output level register m (TOLm)

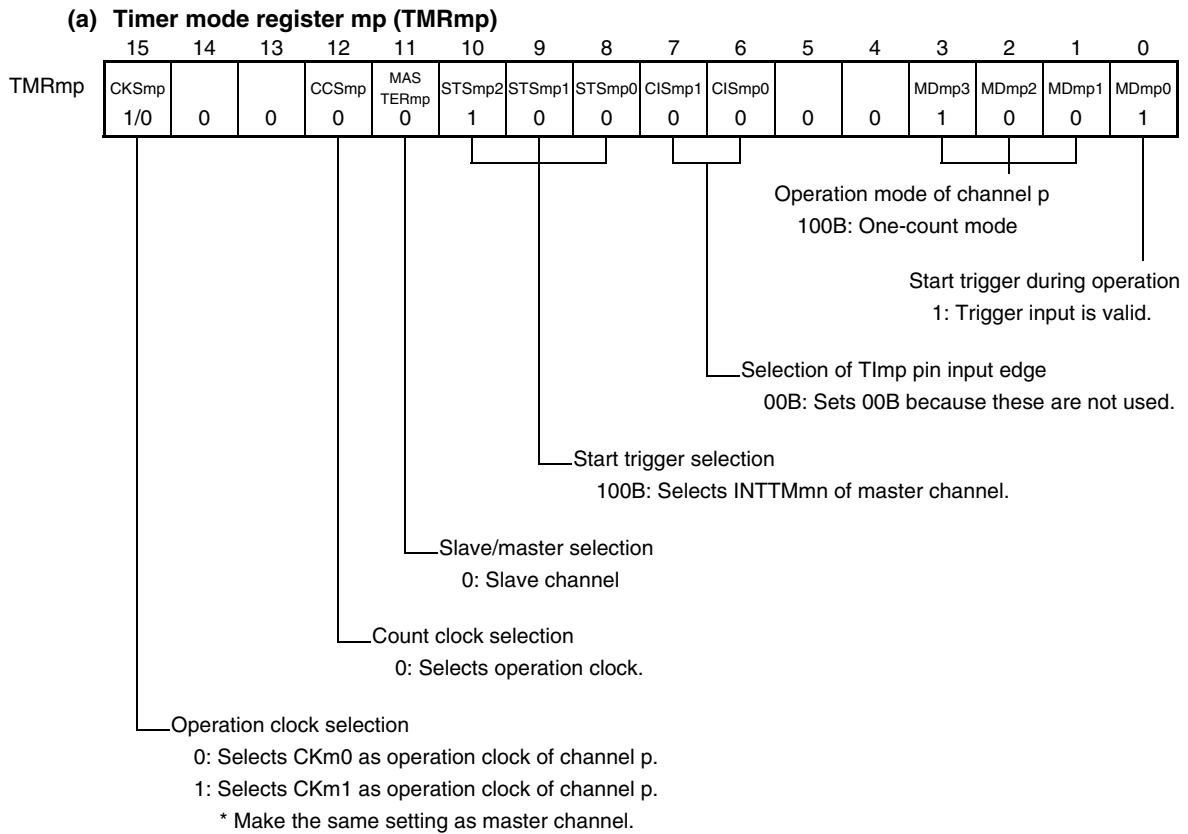


(e) Timer output mode register m (TOMm)

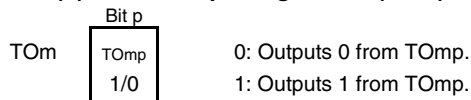


**Remark** m: Unit number, n: Channel number  
 When m = 0: n = 0, 2, 4, 6  
 When m = 1: n = 0, 2

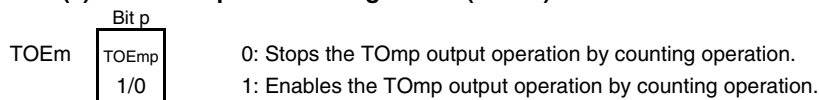
Figure 7-60. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used



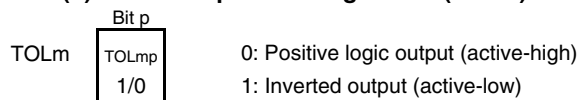
(b) **Timer output register m (TOM)**



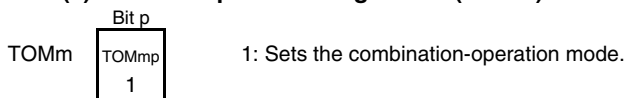
(c) **Timer output enable register m (TOEm)**



(d) **Timer output level register m (TOLm)**



(e) **Timer output mode register m (TOMm)**



**Remark** m: Unit number, n: Channel number, p: Slave channel number (p = n+1)  
 When m = 0: n = 0, 2, 4, 6  
 When m = 1: n = 0, 2

Figure 7-61. 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, TAU1EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the TMRmn and TMRmp registers of two channels to be used (determines operation mode of channels). An interval (period) value is set to the TDRmn register of the master channel, and a duty factor is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of the TOMm register is set to 1 (combination-operation mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp pin goes into Hi-Z output state.
	Sets TOEmp to 1 and enables operation of TOmp. → Clears the port register and port mode register to 0. →	The TOmp default setting level is output when the port mode register is in output mode and the port register is 0. TOmp does not change because channel stops operating. The TOmp pin outputs the TOmp set level.

**Remark** m: Unit number, n: Channel number, p: Slave channel number (p = n+1)  
 When m = 0: n = 0, 2, 4, 6  
 When m = 1: n = 0, 2

Figure 7-61. Operation Procedure When PWM Function Is Used (2/2)

	Software Operation	Hardware Status
Operation is resumed.	<p><b>Operation start</b></p> <p>Sets TOEmp (slave) to 1 (only when operation is resumed).                      The TSmn (master) and TSmp (slave) bits of the TSm register are set to 1 at the same time. →                      The TSmn and TSmp bits automatically return to 0 because they are trigger bits.</p>	<p>TEmn = 1, TEm = 1                      When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.</p>
	<p><b>During operation</b></p> <p>Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed.                      Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated.                      The TCRmn and TCRmp registers can always be read.                      The TSRmn and TSRmp registers are not used.                      Set values of the TOM and TOEm registers can be changed.</p>	<p>The counter of the master channel loads the TDRmn value to TCRmn, and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to TCRmn, and the counter starts counting down again.                      At the slave channel, the value of TDRmp is loaded to TCRmp, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped.                      After that, the above operation is repeated.</p>
	<p><b>Operation stop</b></p> <p>The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. →                      The TTmn and TTmp bits automatically return to 0 because they are trigger bits.</p>	<p>TEmn, TEm = 0, and count operation stops.                      TCRmn and TCRmp hold count value and stops.                      The TOmp output is not initialized but holds current status.</p>
	<p>TOEmp of slave channel is cleared to 0 and value is set to the TOmp bit. →</p>	<p>The TOmp pin outputs the TOMn set level.</p>
<p><b>TAU stop</b></p> <p>To hold the TOmp pin output levels                      Clears TOmp bit to 0 after the value to be held is set to the port register.                      When holding the TOmp pin output levels is not necessary                      Switches the port mode register to input mode. →                      The TAU0EN bit, TAU1ENbit of the PER0 register is cleared to 0. →</p>	<p>The TOmp pin output levels is held by port function.                      The TOmp pin output levels go are into Hi-Z output state.                      Power-off status                      All circuits are initialized and SFR of each channel is also initialized.                      (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)</p>	

**Remark** m: Unit number, n: Channel number, p: Slave channel number (p = n+1)  
 When m = 0: n = 0, 2, 4, 6  
 When m = 1: n = 0, 2

### 7.8.2 Operation as one-shot pulse output function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the signal input to the TImn pin.

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

$$\text{Delay time} = \{\text{Set value of TDRmn (master)} + 2\} \times \text{Count clock period}$$

$$\text{Pulse width} = \{\text{Set value of TDRmp (slave)}\} \times \text{Count clock period}$$

The Master channel operates in the one-count mode and counts the delays. TCRmn of the master channel starts operating upon start trigger detection and TCRmn loads the value of TDRmn. TCRmn counts down from the value of TDRmn it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next start trigger is detected.

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

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

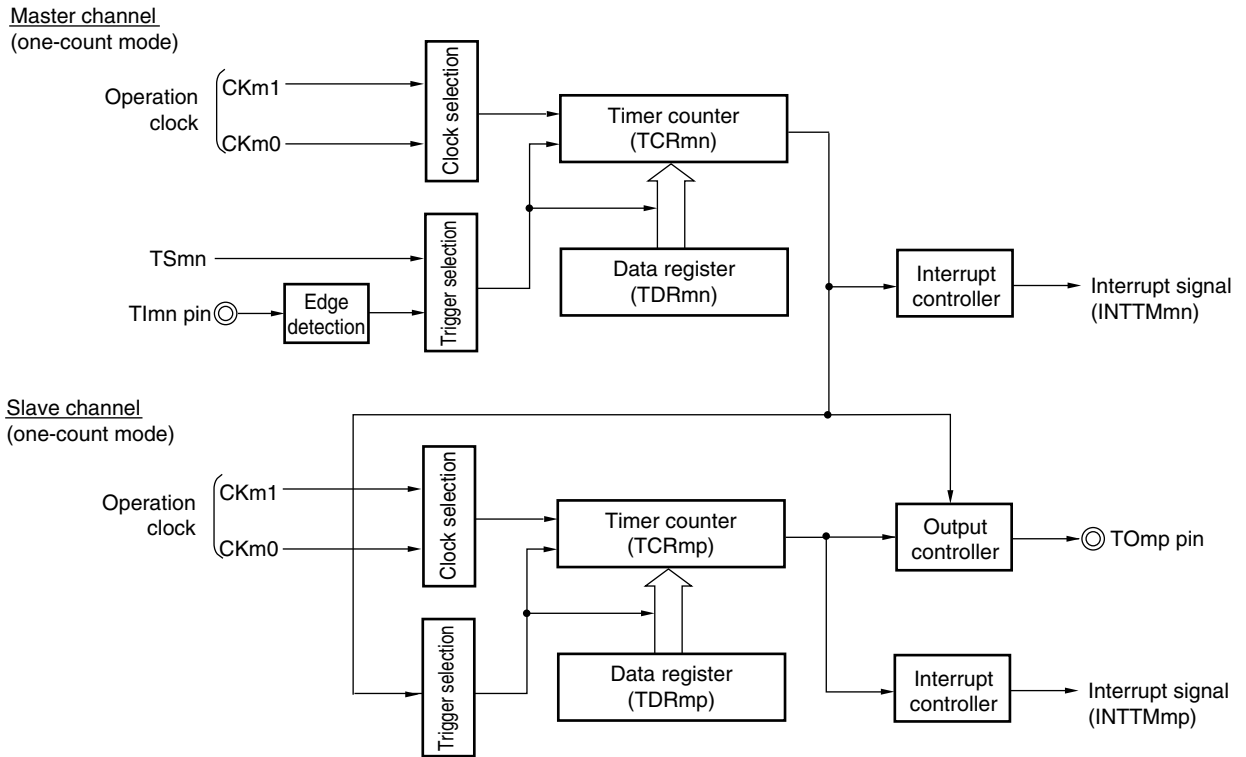
**Caution** The timing of loading of TDRmn of the master channel is different from that of TDRmp of the slave channel. If TDRmn and TDRmp are rewritten during operation, therefore, an illegal waveform is output. Rewrite the TDRmn after INTTMmn is generated and the TDRmp after INTTMmp is generated.

**Remark** m: Unit number, n: Channel number, p: Slave channel number (p = n+1)

When m = 0: n = 0, 2, 4, 6

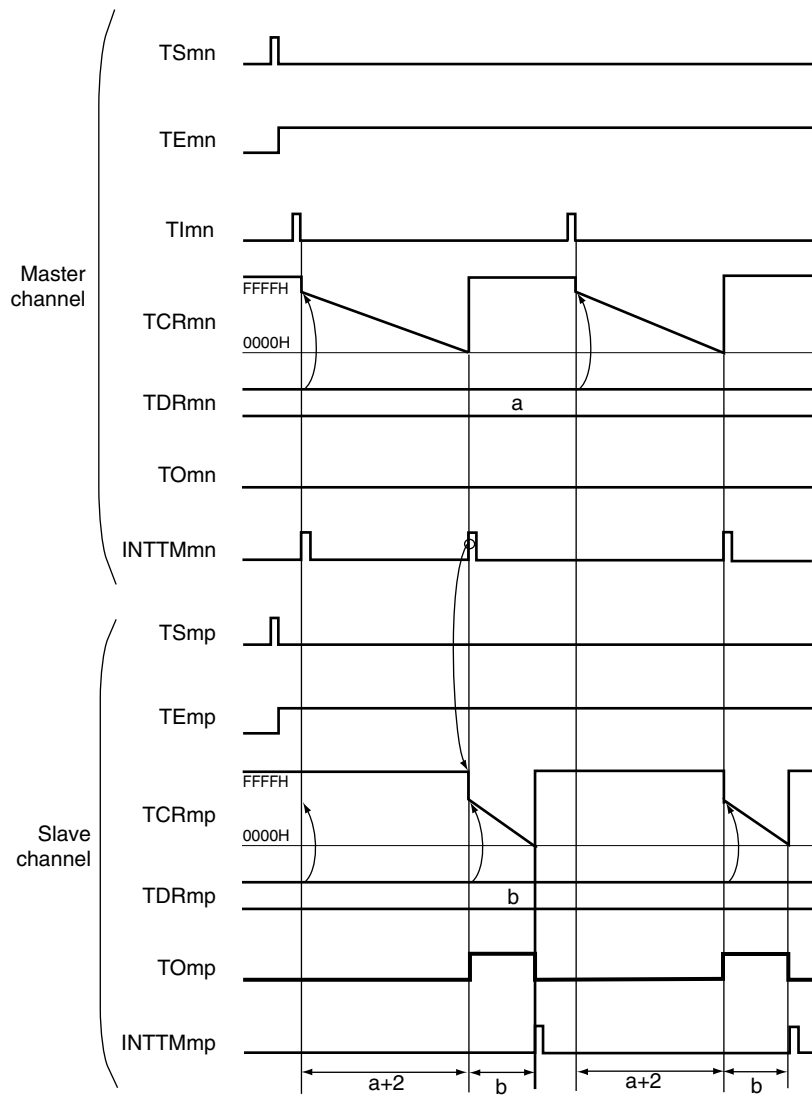
When m = 1: n = 0, 2

Figure 7-62. Block Diagram of Operation as One-Shot Pulse Output Function



**Remark** m: Unit number, n: Channel number, p: Slave channel number (p = n+1)  
 When m = 0: n = 0, 2, 4, 6  
 When m = 1: n = 0, 2

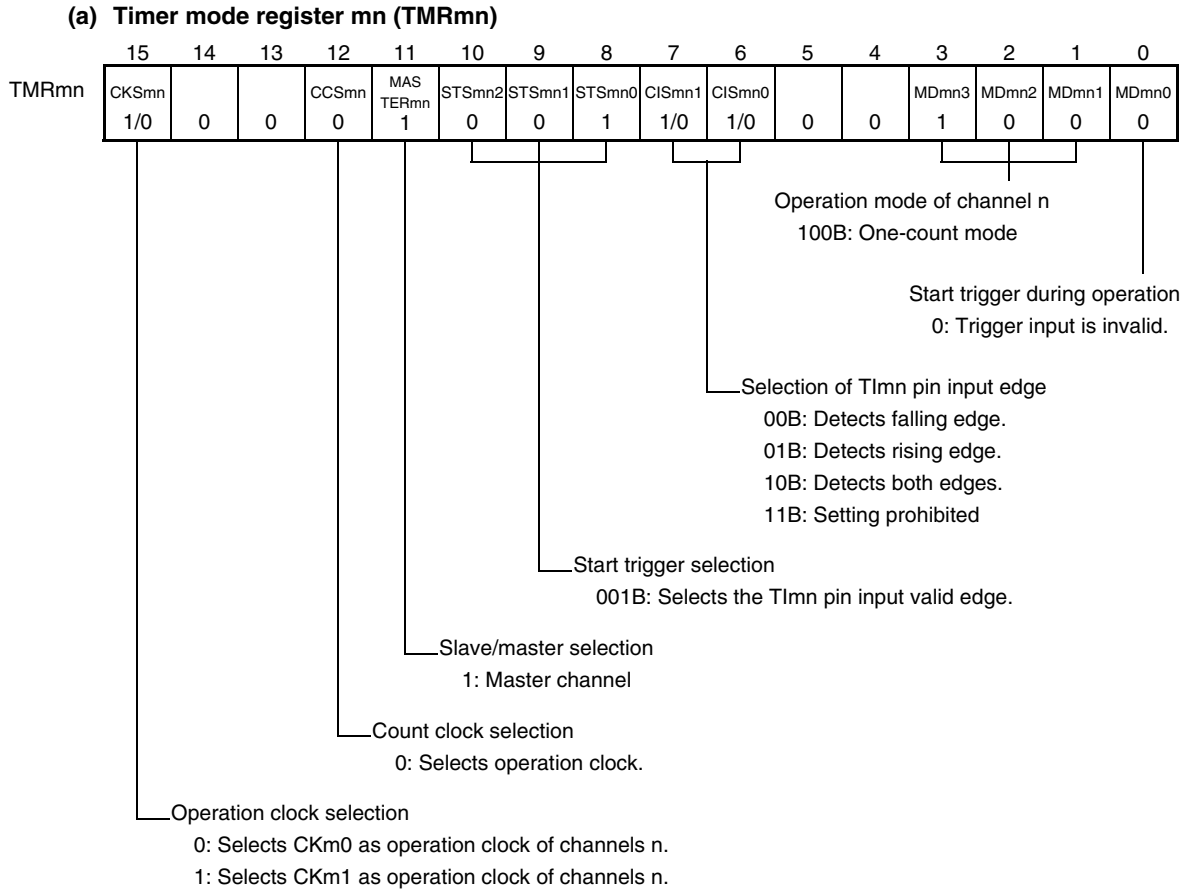
Figure 7-63. Example of Basic Timing of Operation as One-Shot Pulse Output Function



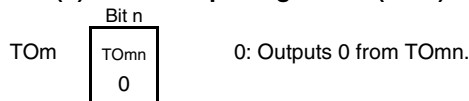
**Remark** m: Unit number, n: Channel number, p: Slave channel number (p = n+1)  
 When m = 0: n = 0, 2, 4, 6  
 When m = 1: n = 0, 2



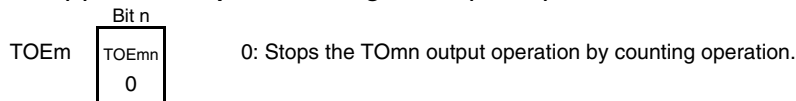
**Figure 7-64. Example of Set Contents of Registers  
When One-Shot Pulse Output Function Is Used (Master Channel)**



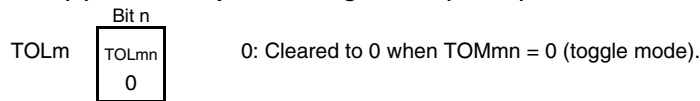
**(b) Timer output register m (TOM)**



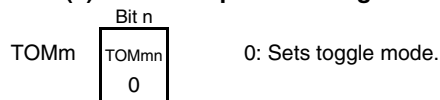
**(c) Timer output enable register m (TOEm)**



**(d) Timer output level register m (TOLm)**

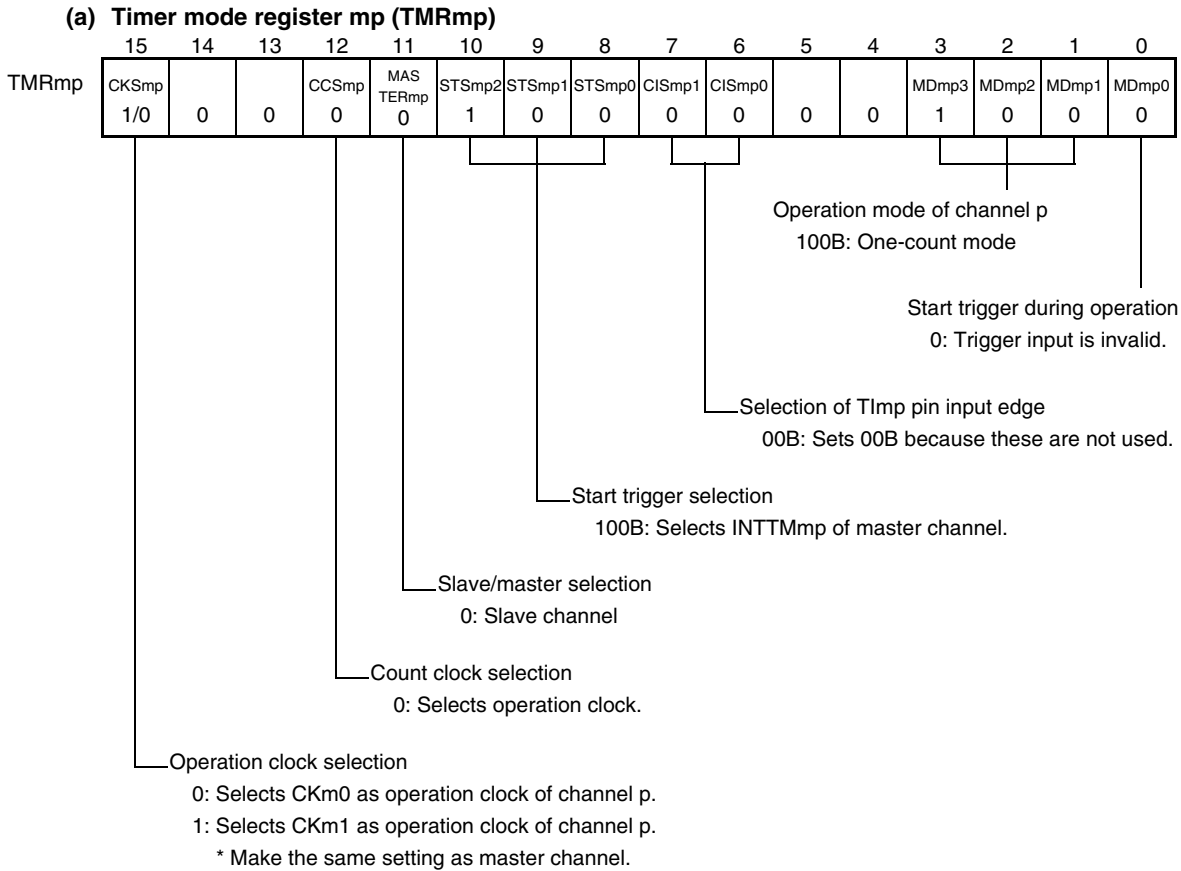


**(e) Timer output mode register m (TOMm)**



**Remark** m: Unit number, n: Channel number, p: Slave channel number (p = n+1),  
 When m = 0: n = 0, 2, 4, 6  
 When m = 1: n = 0, 2

**Figure 7-65. Example of Set Contents of Registers  
When One-Shot Pulse Output Function Is Used (Slave Channel)**



**(b) Timer output register m (TOM)**

	Bit p	
TOM	TOMp	0: Outputs 0 from TOMp.
	1/0	1: Outputs 1 from TOMp.

**(c) Timer output enable register m (TOEm)**

	Bit p	
TOEm	TOEmp	0: Stops the TOMp output operation by counting operation.
	1/0	1: Enables the TOMp output operation by counting operation.

**(d) Timer output level register m (TOLm)**

	Bit p	
TOLm	TOLmp	0: Positive logic output (active-high)
	1/0	1: Inverted output (active-low)

**(e) Timer output mode register m (TOMm)**

	Bit p	
TOMm	TOMmp	1: Sets the combination-operation mode.
	1	

**Remark** m: Unit number, n: Channel number, p: Slave channel number (p = n+1),  
When m = 0: n = 0, 2, 4, 6, When m = 1: n = 0, 2

Figure 7-66. 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, TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the TMRmn and TMRmp registers of two channels to be used (determines operation mode of channels). An output delay is set to the TDRmn register of the master channel, and a pulse width is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel.  The TOMmp bit of the TOMm register is set to 1 (combination-operation mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp pin goes into Hi-Z output state.
	Sets TOEmp to 1 and enables operation of TOmp. → Clears the port register and port mode register to 0. →	The TOmp default setting level is output when the port mode register is in output mode and the port register is 0. TOmp does not change because channel stops operating. The TOmp pin outputs the TOmp set level.

**Remark** m: Unit number, n: Channel number, p: Slave channel number ( $p = n+1$ ),  
 When  $m = 0$ :  $n = 0, 2, 4, 6$   
 When  $m = 1$ :  $n = 0, 2$

Figure 7-66. Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
Operation start	Sets TOEmp (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of the TSm register are set to 1 at the same time. →	TEmn and TEmp are set to 1 and the master channel enters the TImn input edge detection wait status. Counter stops operating.
	The TSmn and TSmp bits automatically return to 0 because they are trigger bits. ----- Detects the TImn pin input valid edge of master channel. →	Master channel starts counting.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOM and TOEm registers can be changed.	Master channel loads the value of TDRmn to TCRmn when the TImn pin valid input edge is detected, and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and the counter stops until the next valid edge is input to the TImn pin. The slave channel, triggered by INTTMmn of the master channel, loads the value of TDRmp to TCRmp, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. →	TEmn, TEmp = 0, and count operation stops. TCRmn and TCRmp hold count value and stops. The TOmp output is not initialized but holds current status.
	The TTmn and TTmp bits automatically return to 0 because they are trigger bits. ----- TOEmp of slave channel is cleared to 0 and value is set to the TOmp bit. →	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output levels Clears TOmp bit to 0 after the value to be held is set to the port register. →	The TOmp pin output levels is held by port function.
	When holding the TOmp pin output levels is not necessary Switches the port mode register to input mode. →	The TOmp pin output levels go are into Hi-Z output state.
	The TAU0EN bit, TAU1EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Operation is resumed.

**Remark** m: Unit number, n: Channel number, p: Slave channel number (p = n+1),  
When m = 0: n = 0, 2, 4, 6  
When m = 1: n = 0, 2

### 7.8.3 Operation as multiple PWM output function

By extending the PWM function and using two or more slave channels, many PWM output signals can be produced.

For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

$$\begin{aligned} \text{Pulse period} &= \{\text{Set value of TDRmn (master)} + 1\} \times \text{Count clock period} \\ \text{Duty factor 1 [\%]} &= \{\text{Set value of TDRmp (slave 1)}\} / \{\text{Set value of TDRmn (master)} + 1\} \times 100 \\ \text{Duty factor 2 [\%]} &= \{\text{Set value of TDRmq (slave 2)}\} / \{\text{Set value of TDRmn (master)} + 1\} \times 100 \end{aligned}$$

**Remark** Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

TCRmn of the master channel operates in the interval timer mode and counts the periods.

TCRmp of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. TCRmp loads the value of TDRmp to TCRmp, using INTTMmn of the master channel as a start trigger, and start counting down. When TCRmp = 0000H, TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

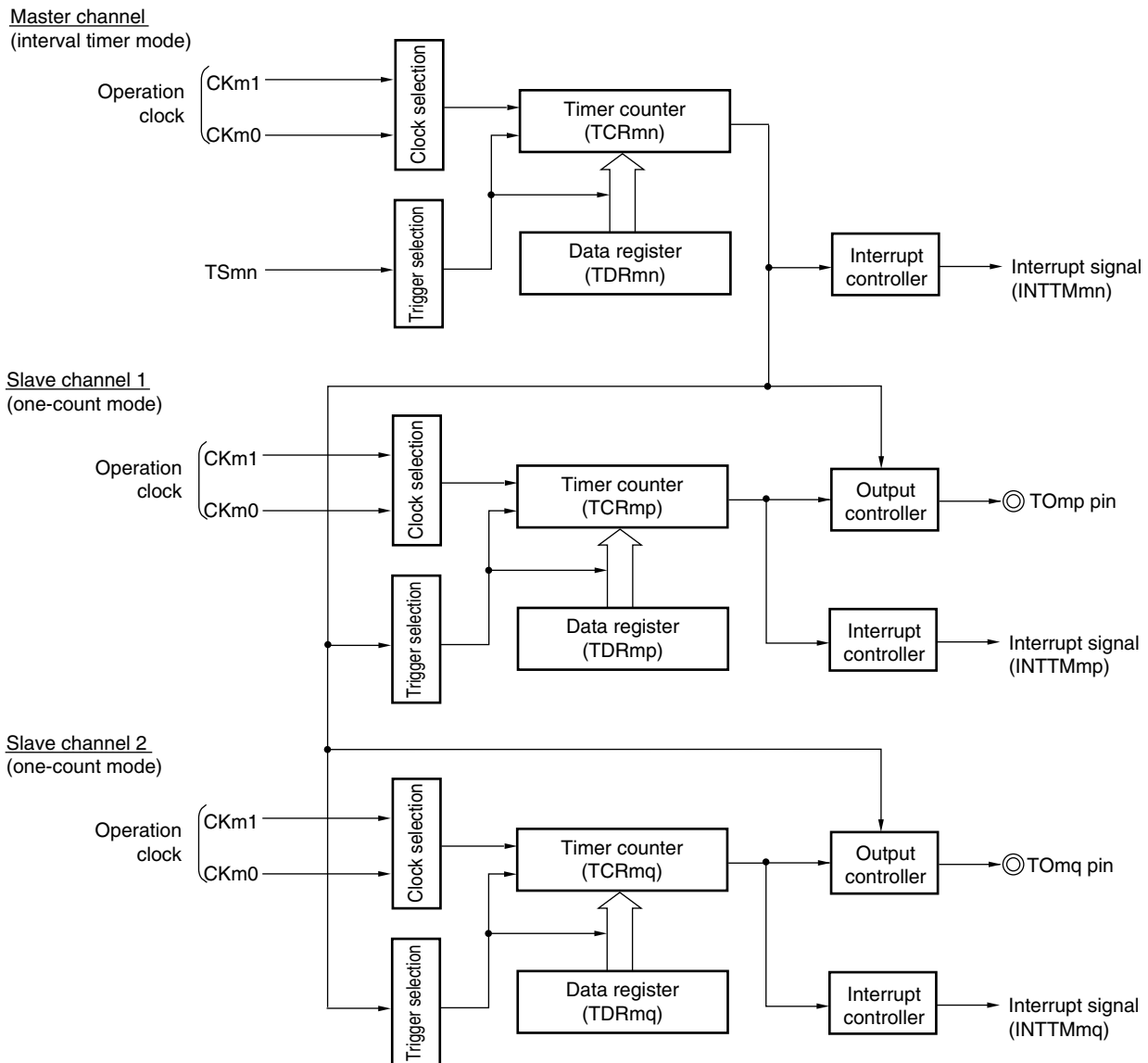
In the same way as TCRmp of the slave channel 1, TCRmq of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOMq pin. TCRmq loads the value of TDRmq to TCRmq, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = 0000H, TCRmq outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOMq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.

When channel 0 is used as the master channel as above, as for the timer array unit 0, up to seven types of PWM signals can be generated, while as for the timer array unit 1, up to three types of PWM signals can be generated.

**Caution** To rewrite both TDRmn of the master channel and TDRmp of the slave channel 1, write access is necessary at least twice. Since the values of TDRmn and TDRmp are loaded to TCRmn and TCRmp after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both TDRmn of the master and TDRmp of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to TDRmq of the slave channel 2).

**Remark** m: Unit number, n: Channel number, p: Slave channel number 1 (n+1), q: Slave channel number 2 (n+2)  
 When m = 0  
     n = 0, 2, 4  
     n < p < q ≤ 7 (where p and q are a consecutive integer greater than n)  
 When m = 1  
     n = 0  
     n < p < q ≤ 3 (where p and q are a consecutive integer greater than n (p = 1, q = 2))

Figure 7-67. Block Diagram of Operation as Multiple PWM Output Function (output two types of PWMs)



**Remark** m: Unit number, n: Channel number, p: Slave channel number 1 (n+1), q: Slave channel number 2 (n+2)

When m = 0

n = 0, 2, 4

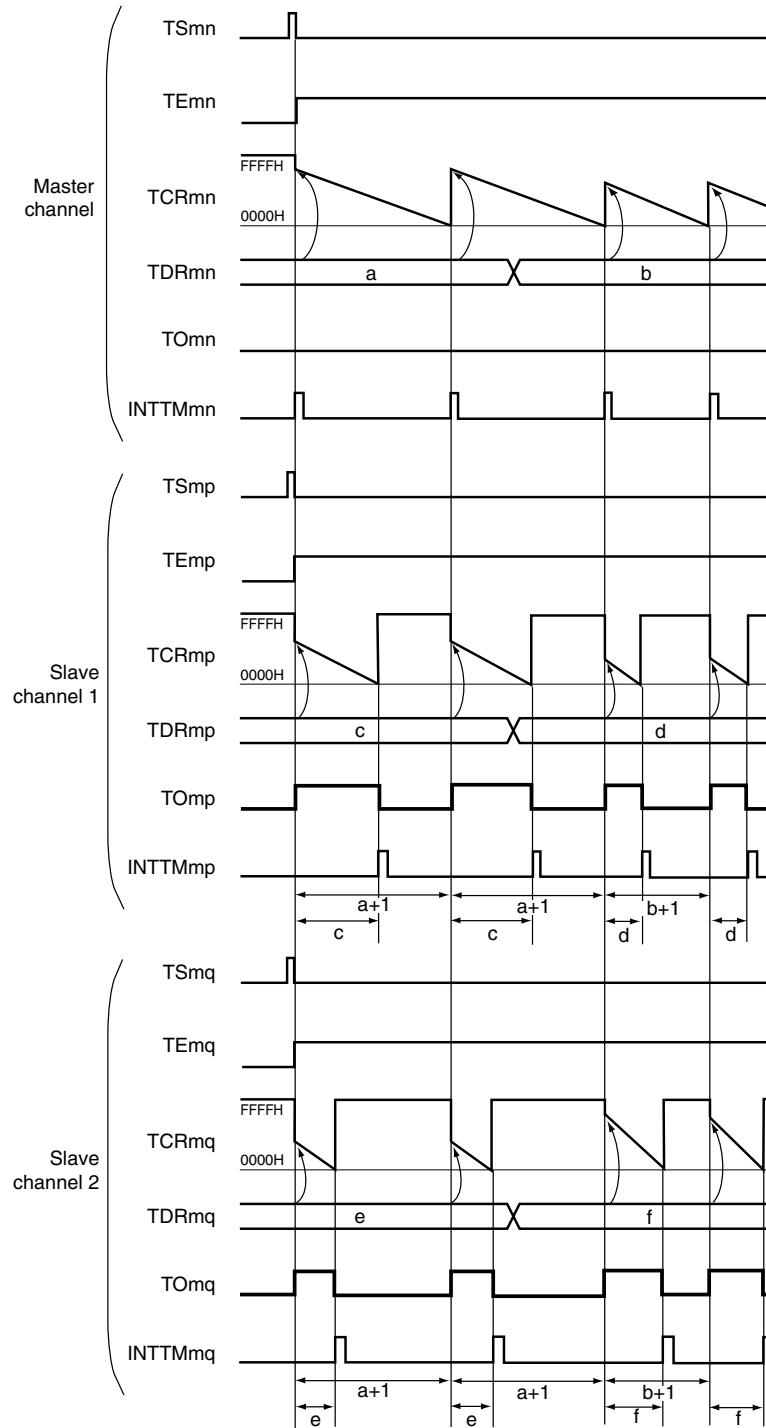
n < p < q ≤ 7 (where p and q are a consecutive integer greater than n)

When m = 1

n = 0

n < p < q ≤ 3 (where p and q are a consecutive integer greater than n ( p = 1, q = 2))

Figure 7-68. Example of Basic Timing of Operation as Multiple PWM Output Function (output two types of PWMs)

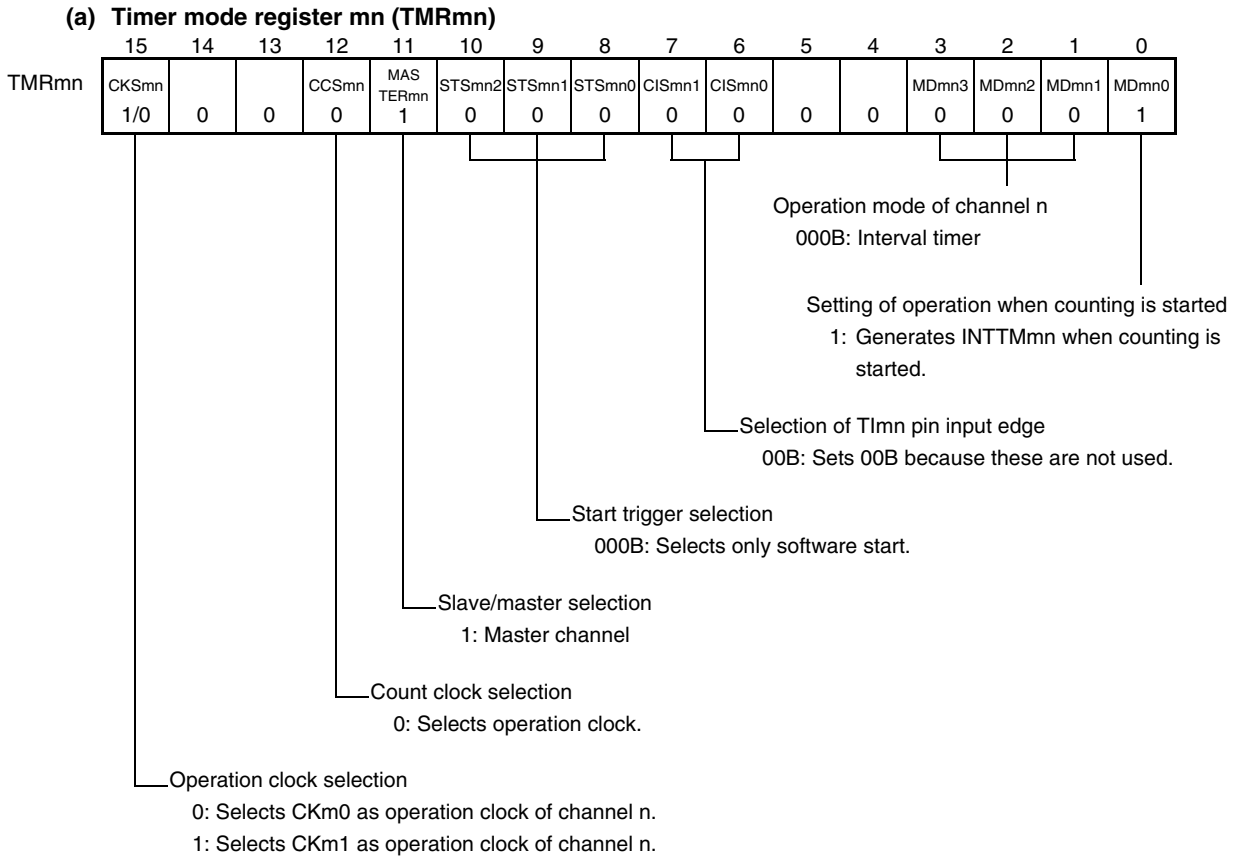


**Remark** m: Unit number, n: Channel number, p: Slave channel number 1 (n+1), q: Slave channel number 2 (n+2)

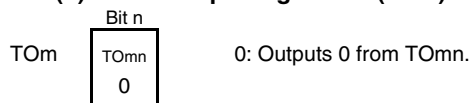
When m = 0  
 n = 0, 2, 4  
 n < p < q ≤ 7 (where p and q are a consecutive integer greater than n)

When m = 1  
 n = 0  
 n < p < q ≤ 3 (where p and q are a consecutive integer greater than n ( p = 1, q = 2))

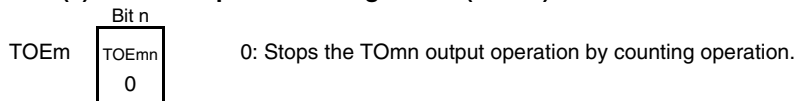
**Figure 7-69. Example of Set Contents of Registers  
When Multiple PWM Output Function (Master Channel) Is Used**



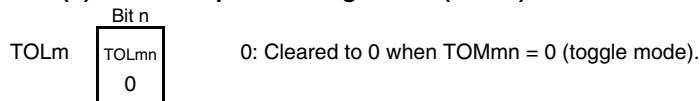
**(b) Timer output register m (TOM)**



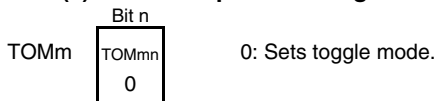
**(c) Timer output enable register m (TOEm)**



**(d) Timer output level register m (TOLm)**



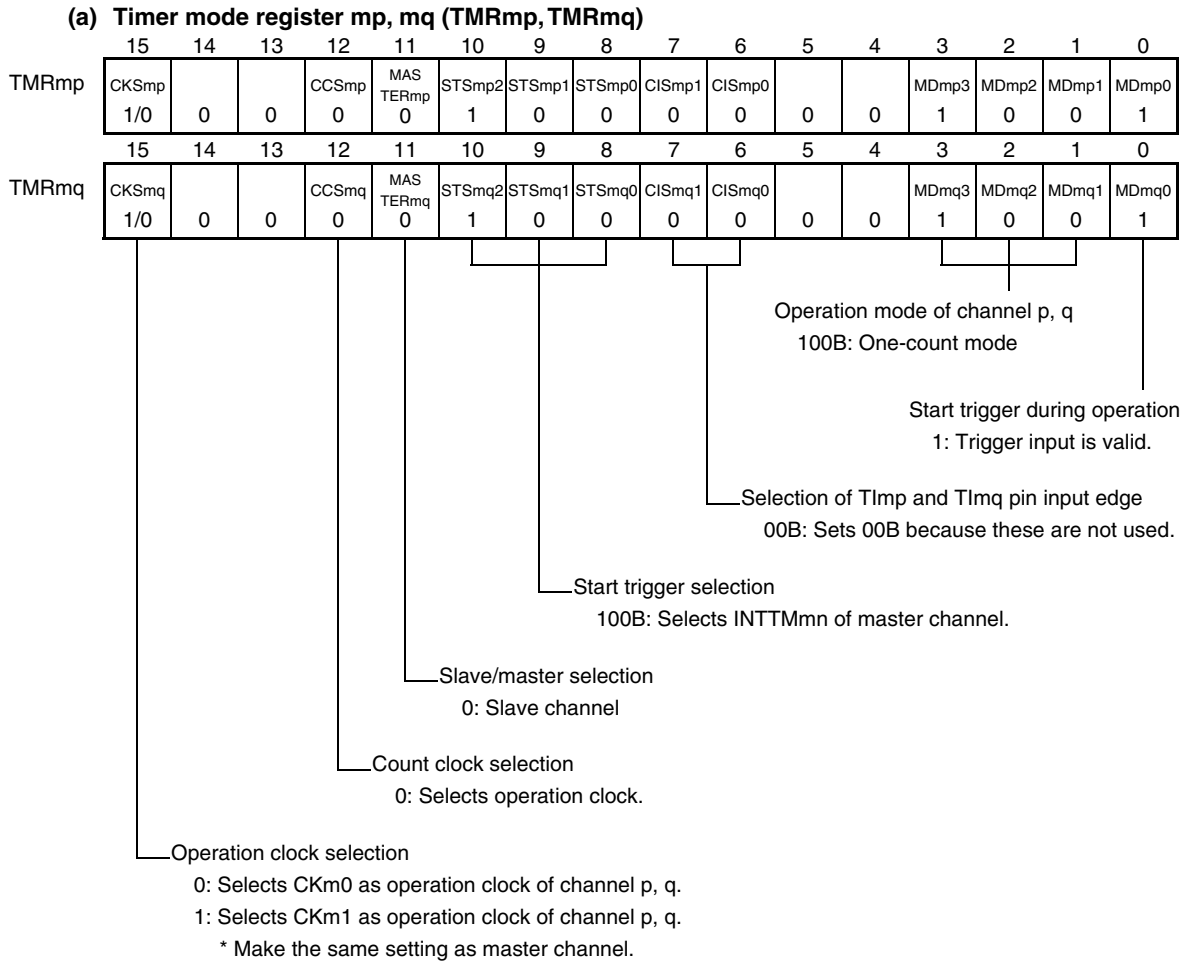
**(e) Timer output mode register m (TOMm)**



**Remark** m: Unit number (m = 0, 1), n: Channel number  
 When m = 0: n = 0, 2, 4  
 When m = 1: n = 0



**Figure 7-70. Example of Set Contents of Registers**  
**When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)**



**(b) Timer output register m (TOm)**

	Bit q	Bit p	
TOm	TOmq	TOmp	0: Outputs 0 from TOmp or TOmq. 1: Outputs 1 from TOmp or TOmq.
	1/0	1/0	

**(c) Timer output enable register m (TOEm)**

	Bit q	Bit p	
TOEm	TOEmq	TOEmp	0: Stops the TOmp or TOmq output operation by counting operation. 1: Enables the TOmp or TOmq output operation by counting operation.
	1/0	1/0	

**(d) Timer output level register m (TOLm)**

	Bit q	Bit p	
TOLm	TOLmq	TOLmp	0: Positive logic output (active-high) 1: Inverted output (active-low)
	1/0	1/0	

**(e) Timer output mode register m (TOMm)**

	Bit q	Bit p	
TOMm	TOMmq	TOMmp	1: Sets the combination-operation mode.
	1	1	

(Remark are given on the next page.)

**Remark** m: Unit number, n: Channel number, p: Slave channel number 1 (n+1), q: Slave channel number 2 (n+2)

When m = 0

n = 0, 2, 4

n < p < q ≤ 7 (where p and q are a consecutive integer greater than n)

When m = 1

n = 0

n < p < q ≤ 3 (where p and q are a consecutive integer greater than n ( p = 1, q = 2))

Figure 7-71. Operation Procedure When Multiple PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit, TAU1EN bit of the PER0 register to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the TMRmn, TMRmp, and TMRmq registers of each channel to be used (determines operation mode of channels). An interval (period) value is set to the TDRmn register of the master channel, and a duty factor is set to the TDRmp and TDRmq registers of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp and TOMmq bits of the TOMm register is set to 1 (combination-operation mode). Clears the TOLmp and TOLmq bits to 0. Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs.	The TOmp and TOmq pins go into Hi-Z output state.
	Sets TOEmp and TOEmq to 1 and enables operation of TOmp and TOmq.	The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Clears the port register and port mode register to 0.	TOmp or TOmq does not change because channel stops operating. The TOmp and TOmq pins output the TOmp and TOmq set levels.
Operation start	Sets TOEmp and TOEmq (slave) to 1 (only when operation is resumed). The TSmn bit (master), and TSmp and TSmq (slave) bits of the TSm register are set to 1 at the same time. The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmq = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.

Operation is resumed (on the next page).

**Remark** m: Unit number(m = 0, 1), n: Channel number, p: Slave channel number 1 (n+1)  
q: Slave channel number 2 (n+2)

When m = 0

n = 0, 2, 4

n < p < q ≤ 7 (where p and q are a consecutive integer greater than n)

When m = 1

n = 0

n < p < q ≤ 3 (where p and q are a consecutive integer greater than n ( p = 1, q = 2))

Figure 7-71. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

Operation is resumed (on the before page).

	Software Operation	Hardware Status
During operation	<p>Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed.</p> <p>Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated.</p> <p>The TCRmn, TCRmp, and TCRmq registers can always be read.</p> <p>The TSRmn, TSRmp, and TSRmq registers are not used.</p> <p>Set values of the TOm and TOEm registers can be changed.</p>	<p>The counter of the master channel loads the TDRmn value to TCRmn and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated.</p> <p>At the same time, the value of the TDRmn register is loaded to TCRmn, and the counter starts counting down again.</p> <p>At the slave channel 1, the values of TDRmp are transferred to TCRmp, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOMp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped.</p> <p>At the slave channel 2, the values of TDRmq are transferred to TCRmq, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOMq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped.</p> <p>After that, the above operation is repeated.</p>
Operation stop	<p>The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. —————&gt;</p> <p>The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.</p>	<p>TEmn, TEmp and TEmq = 0, and count operation stops. TCRmn, TCRmp and TCRmq hold count value and stops.</p> <p>The TOMp and TOMq output is not initialized but holds current status.</p>
	<p>TOEmp or TOEmq of slave channel is cleared to 0 and value is set to the TOMp and TOMq bits. —————&gt;</p>	<p>The TOMp and TOMq pins output the TOMp and TOMq set levels.</p>
TAU stop	<p>To hold the TOMp and TOMq pin output levels Clears TOMp and TOMq bits to 0 after the value to be held is set to the port register. —————&gt;</p> <p>When holding the TOMp and TOMq pin output levels is not necessary Switches the port mode register to input mode. —————&gt;</p>	<p>The TOMp and TOMq pin output levels are held by port function.</p> <p>The TOMp and TOMq pin output levels go into Hi-Z output state.</p>
	<p>The TAU0EN bit, TAU1EN bit of the PER0 register is cleared to 0. —————&gt;</p>	<p>Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOMp and TOMq bits are cleared to 0 and the TOMp and TOMq pins are set to port mode.)</p>

**Remark** m: Unit number, n: Channel number, p: Slave channel number 1 (n+1), q: Slave channel number 2 (n+2)

When m = 0  
 n = 0, 2, 4  
 n < p < q ≤ 7 (where p and q are a consecutive integer greater than n)

When m = 1  
 n = 0  
 n < p < q ≤ 3 (where p and q are a consecutive integer greater than n ( p = 1, q = 2))

## CHAPTER 8 REAL-TIME COUNTER

### 8.1 Functions of Real-Time Counter

The real-time counter has the following features.

- Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
- Constant-period interrupt function (period: 1 month to 0.5 seconds)
- Alarm interrupt function (alarm: week, hour, minute)
- Interval interrupt function
- Pin output function of 1 Hz
- Pin output function of 512 Hz or 16.384 kHz or 32.768 kHz

### 8.2 Configuration of Real-Time Counter

The real-time counter includes the following hardware.

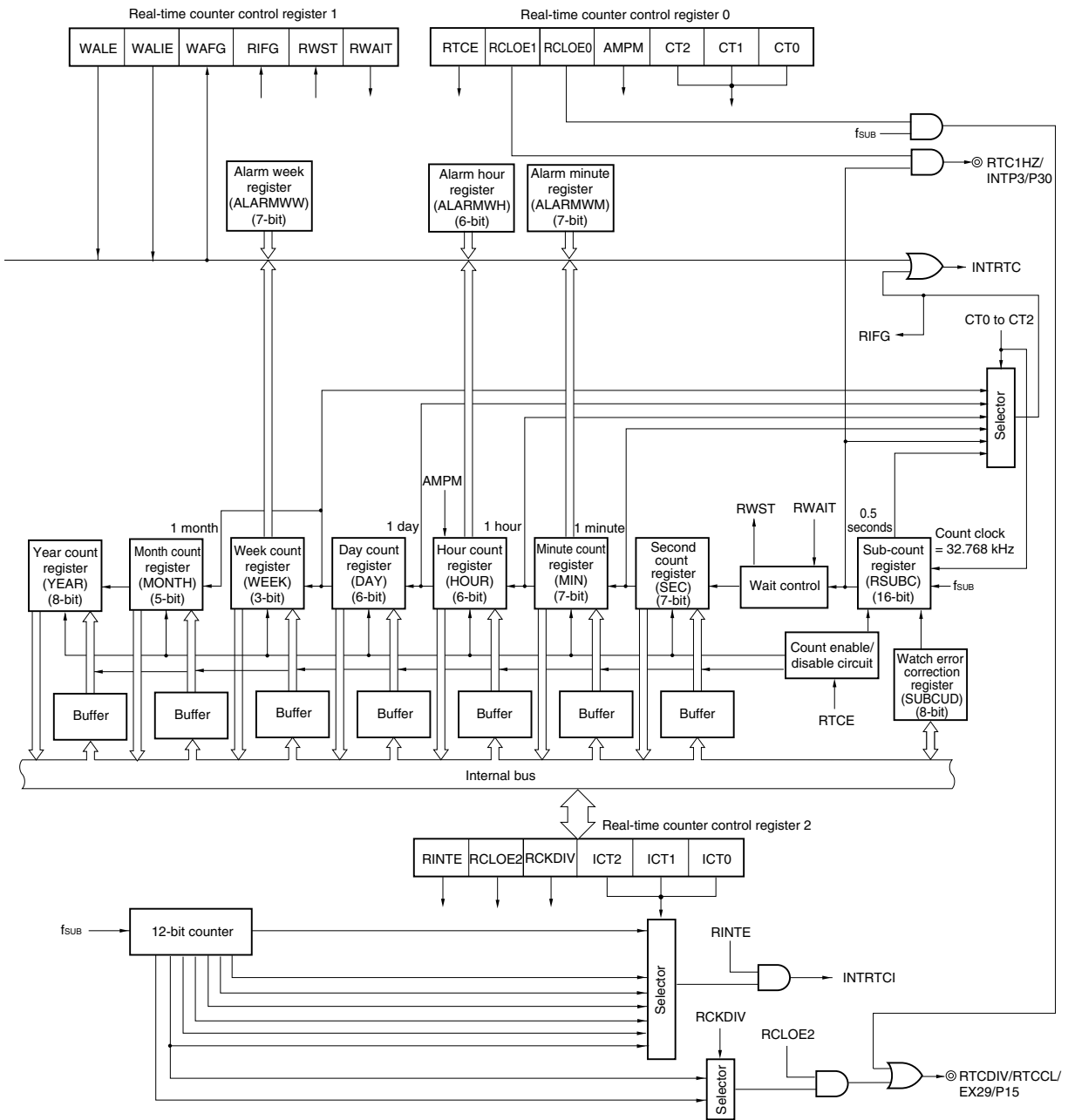
**Table 8-1. Configuration of Real-Time Counter**

Item	Configuration
Control registers	Peripheral enable register 0 (PER0) Real-time counter control register 0 (RTCC0) Real-time counter control register 1 (RTCC1) Real-time counter control register 2 (RTCC2) Sub-count register (RSUBC) Second count register (SEC) Minute count register (MIN) Hour count register (HOUR) Day count register (DAY) Week count register (WEEK) Month count register (MONTH) Year count register (YEAR) Watch error correction register (SUBCUD) Alarm minute register (ALARMWM) Alarm hour register (ALARMWH) Alarm week register (ALARMWW) Port mode registers 1 and 3 (PM1, PM3) Port registers 1 and 3 (P1, P3)

<R>

<R>

Figure 8-1. Block Diagram of Real-Time Counter



### 8.3 Registers Controlling Real-Time Counter

The real-time counter is controlled by the following 18 registers.

- Peripheral enable register 0 (PER0)
- Real-time counter control register 0 (RTCC0)
- Real-time counter control register 1 (RTCC1)
- Real-time counter control register 2 (RTCC2)
- Sub-count register (RSUBC)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Watch error correction register (SUBCUD)
- Alarm minute register (ALARMWMM)
- Alarm hour register (ALARMWH)
- Alarm week register (ALARMWW)
- <R> • Port mode registers 1, 3 (PM1, PM3)
- <R> • Port registers 1, 3 (P1, P3)

**(1) Peripheral enable register 0 (PER0)**

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the real-time counter is used, be sure to set bit 7 (RTCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

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

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of real-time counter (RTC) input clock supply <sup>Note</sup>
0	Stops supply of input clock. <ul style="list-style-type: none"> <li>• SFR used by the real-time counter (RTC) cannot be written.</li> <li>• The real-time counter (RTC) is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the real-time counter (RTC) can be read/written.</li> </ul>

**Note** RTCEN is used to supply or stop the clock used when accessing the real-time counter (RTC) register from the CPU. RTCEN cannot control supply of the operating clock ( $f_{SUB}$ ) to RTC.

**Caution** When using the real-time counter, first set RTCEN to 1, while oscillation of the subsystem clock ( $f_{SUB}$ ) is stable. If RTCEN = 0, writing to a control register of the real-time counter is ignored, and, even if the register is read, only the default value is read.

**(2) Real-time counter control register 0 (RTCC0)**

The RTCC0 register is an 8-bit register that is used to start or stop the real-time counter operation, control the RTCCL and RTC1HZ pins, and set a 12- or 24-hour system and the constant-period interrupt function.

RTCC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.



Figure 8-3. Format of Real-Time Counter Control Register 0 (RTCC0)

Address: FFF9DH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
RTCC0	RTCE	0	RCLOE1	RCLOE0	AMPM	CT2	CT1	CT0

RTCE	Real-time counter operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1	RTC1HZ pin output control
0	Disables output of RTC1HZ pin (1 Hz).
1	Enables output of RTC1HZ pin (1 Hz).

RCLOE0 <sup>Note</sup>	RTCCL pin output control
0	Disables output of RTCCL pin (32.768 kHz).
1	Enables output of RTCCL pin (32.768 kHz).

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system

Rewrite the AMPM value after setting RWAIT (bit 0 of RTCC1) to 1. If the AMPM value is changed, the values of the hour count register (HOUR) change according to the specified time system. Table 8-2 shows the displayed time digits.

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)

When changing the values of CT2 to CT0 while the counter operates (RTCE = 1), rewrite the values of CT2 to CT0 after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of CT2 to CT0, enable interrupt servicing after clearing the RIFG and RTCIF flags.

**Note** RCLOE0 and RCLOE2 must not be enabled at the same time.

**Caution** If RCLOE0 and RCLOE1 are changed when RTCE = 1, glitches may occur in the 32.768 kHz and 1 Hz output signals.

**Remark** ×: don't care

<R>

**(3) Real-time counter control register 1 (RTCC1)**

The RTCC1 register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.

RTCC1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 8-4. Format of Real-Time Counter Control Register 1 (RTCC1) (1/2)**

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
When setting a value to the WALE bit while the counter operates (RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting each alarm register (WALIE flag of RTCC1, the ALARMWM register, the ALARMWH register, and the ALARMWW register), set match operation to be invalid ("0") for the WALE bit.	

WALIE	Control of alarm interrupt (INTRTC) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to "1" one clock (32.768 kHz) after matching of the alarm is detected. This flag is cleared when "0" is written to it. Writing "1" to it is invalid.	

**Figure 8-4. Format of Real-Time Counter Control Register 1 (RTCC1) (2/2)**

RIFG	Constant-period interrupt status flag
0	Constant-period interrupt is not generated.
1	Constant-period interrupt is generated.

This flag indicates the status of generation of the constant-period interrupt. When the constant-period interrupt is generated, it is set to "1".  
This flag is cleared when "0" is written to it. Writing "1" to it is invalid.

RWST	Wait status flag of real-time counter
0	Counter is operating.
1	Mode to read or write counter value

This status flag indicates whether the setting of RWAIT is valid.  
Before reading or writing the counter value, confirm that the value of this flag is 1.

RWAIT	Wait control of real-time counter
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value

This bit controls the operation of the counter.  
Be sure to write "1" to it to read or write the counter value.  
Because RSUBC continues operation, complete reading or writing of it in 1 second, and clear this bit back to 0.  
When RWAIT = 1, it takes up to 1 clock (32.768 kHz) until the counter value can be read or written.  
If RSUBC overflows when RWAIT = 1, it counts up after RWAIT = 0. If the second count register is written, however, it does not count up because RSUBC is cleared.

**Caution** The RIFG and WAFG flags may be cleared when the RTCC1 register is written by using a 1-bit manipulation instruction. Use, therefore, an 8-bit manipulation instruction in order to write to the RTCC1 register. To prevent the RIFG and WAFG flags from being cleared during writing, disable writing by setting "1" to the corresponding bit. When the value may be rewritten because the RIFG and WAFG flags are not being used, the RTCC1 register may be written by using a 1-bit manipulation instruction.

**Remark** Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

**(4) Real-time counter control register 2 (RTCC2)**

The RTCC2 register is an 8-bit register that is used to control the interval interrupt function and the RTCDIV pin.

RTCC2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 8-5. Format of Real-Time Counter Control Register 2 (RTCC2)**

Address: FFF9FH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
RTCC2	RINTE	RCLOE2	RCKDIV	0	0	ICT2	ICT1	ICT0

RINTE	ICT2	ICT1	ICT0	Interval interrupt (INTRTCI) selection
0	×	×	×	Interval interrupt is not generated.
1	0	0	0	$2^6/f_{XT}$ (1.953125 ms)
1	0	0	1	$2^7/f_{XT}$ (3.90625 ms)
1	0	1	0	$2^8/f_{XT}$ (7.8125 ms)
1	0	1	1	$2^9/f_{XT}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{XT}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{XT}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{XT}$ (125 ms)

RCLOE2 <sup>Note</sup>	RTCDIV pin output control
0	Output of RTCDIV pin is disabled.
1	Output of RTCDIV pin is enabled.

RCKDIV	Selection of RTCDIV pin output frequency
0	RTCDIV pin outputs 512 Hz. (1.95 ms)
1	RTCDIV pin outputs 16.384 kHz. (0.061 ms)

**Notes** RCLOE0 and RCLOE2 must not be enabled at the same time.

**Cautions** 1. Change ICT2, ICT1, and ICT0 when RINTE = 0.

2. When the output from RTCDIV pin is stopped, the output continues after a maximum of two clocks of  $f_{XT}$  and enters the low level. While 512 Hz is output, and when the output is stopped immediately after entering the high level, a pulse of at least one clock width of  $f_{XT}$  may be generated.

3. After the real-time counter starts operating, the output width of the RTCDIV pin may be shorter than as set during the first interval period.

**(5) Sub-count register (RSUBC)**

The RSUBC register is a 16-bit register that counts the reference time of 1 second of the real-time counter. It takes a value of 0000H to 7FFFH and counts 1 second with a clock of 32.768 kHz.

RSUBC can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

- Cautions 1.** When a correction is made by using the SUBCUD register, the value may become 8000H or more.
- 2.** This register is also cleared by reset effected by writing the second count register.
- 3.** The value read from this register is not guaranteed if it is read during operation, because a value that is changing is read.

**Figure 8-6. Format of Sub-Count Register (RSUBC)**

Address: FFF90H After reset: 0000H R

Symbol	7	6	5	4	3	2	1	0
RSUBC	SUBC7	SUBC6	SUBC5	SUBC4	SUBC3	SUBC2	SUBC1	SUBC0

Address: FFF91H After reset: 0000H R

Symbol	7	6	5	4	3	2	1	0
RSUBC	SUBC15	SUBC14	SUBC13	SUBC12	SUBC11	SUBC10	SUBC9	SUBC8

**(6) Second count register (SEC)**

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the sub-counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 00H.

**Figure 8-7. Format of Second Count Register (SEC)**

Address: FFF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

**(7) Minute count register (MIN)**

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Even if the second count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

<R>

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

Reset signal generation clears this register to 00H.

**Figure 8-8. Format of Minute Count Register (MIN)**

Address: FFF93H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

**(8) Hour count register (HOUR)**

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12, 21 to 32 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using bit 3 (AMPM) of real-time counter control register 0 (RTCC0).

<R>

If the AMPM bit value is changed, the values of the HOUR register change according to the specified time system.

If a value outside the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

**Figure 8-9. Format of Hour Count Register (HOUR)**

Address: FFF94H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

**Caution Bit 5 (HOUR20) of HOUR indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).**

**Table 8-2. Displayed Time Digits**

24-Hour Display (AMPM Bit = 1)		12-Hour Display (AMPM Bit = 0)	
Time	HOUR Register	Time	HOUR Register
0	00H	0 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	0 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

**(9) Day count register (DAY)**

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.

It counts up when the hour counter overflows.

This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Even if the hour count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 31 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 01H.

**Figure 8-10. Format of Day Count Register (DAY)**

Address: FFF96H    After reset: 01H    R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1



**(10) Week count register (WEEK)**

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.

It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 06 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 00H.

**Figure 8-11. Format of Week Count Register (WEEK)**

Address: FFF95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

**Caution** The value corresponding to the month count register or the day count register is not stored in the week count register automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

**(11) Month count register (MONTH)**

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 01H.

**Figure 8-12. Format of Month Count Register (MONTH)**

Address: FFF97H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

**(12) Year count register (YEAR)**

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month counter overflows.

Values 00, 04, 08, ..., 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Even if the month count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code. If a value outside the range is set, the register value returns to the normal value after 1 period.

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

Reset signal generation clears this register to 00H.

**Figure 8-13. Format of Year Count Register (YEAR)**

Address: FFF98H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

**(13) Watch error correction register (SUBCUD)**

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value (reference value: 7FFFH) that overflows from the sub-count register (RSUBC) to the second count register. SUBCUD can be set by an 8-bit memory manipulation instruction. Reset signal generation clears this register to 00H.

**Figure 8-14. Format of Watch Error Correction Register (SUBCUD)**

Address: FFF99H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch error only when the second digits are at 00 (every 60 seconds).
Writing to the SUBCUD register at the following timing is prohibited.	
<ul style="list-style-type: none"> <li>• When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H</li> <li>• When DEV = 1 is set: For a period of SEC = 00H</li> </ul>	

F6	Setting of watch error correction value
0	Increases by $\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2$ .
1	Decreases by $\{(\text{/F5, /F4, /F3, /F2, /F1, /F0}) + 1\} \times 2$ .
When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected. * is 0 or 1. /F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100).	
Range of correction value: (when F6 = 0) 2, 4, 6, 8, ..., 120, 122, 124	
(when F6 = 1) -2, -4, -6, -8, ..., -120, -122, -124	

<R>

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	-189.2 ppm to 189.2 ppm	-63.1 ppm to 63.1 ppm
Maximum excludes quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

**Remark** Set DEV to 0 when the correction range is -63.1 ppm or less, or 63.1 ppm or more.

**(14) Alarm minute register (ALARMWWM)**

This register is used to set minutes of alarm.

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

Reset signal generation clears this register to 00H.

**Caution** Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

**Figure 8-15. Format of Alarm Minute Register (ALARMWWM)**

Address: FFF9AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

**(15) Alarm hour register (ALARMWH)**

This register is used to set hours of alarm.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

**Caution** Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

**Figure 8-16. Format of Alarm Hour Register (ALARMWH)**

Address: FFF9BH After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

**Caution** Bit 5 (WH20) of ALARMWH indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

**(16) Alarm week register (ALARMWW)**

This register is used to set date of alarm.

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

Reset signal generation clears this register to 00H.

**Figure 8-17. Format of Alarm Week Register (ALARMWW)**

Address: FFF9CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
	W W 0	W W 1	W W 2	W W 3	W W 4	W W 5	W W 6								
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

<R>

**(17) Port mode registers 1, 3 (PM1, PM3)**

This register sets ports 1 and 3 input/output in 1-bit units.

When using the P15/EX29/RTCDIV/RTCCL and P30/RTC1HZ/INTP3 pins for clock output of real-time counter, clear PM15 and PM30 and the output latches of P15 and P30 to 0.

PM1 and PM3 are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

**Figure 8-18. Format of Port Mode Registers 1 and 3 (PM1, PM3)**

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

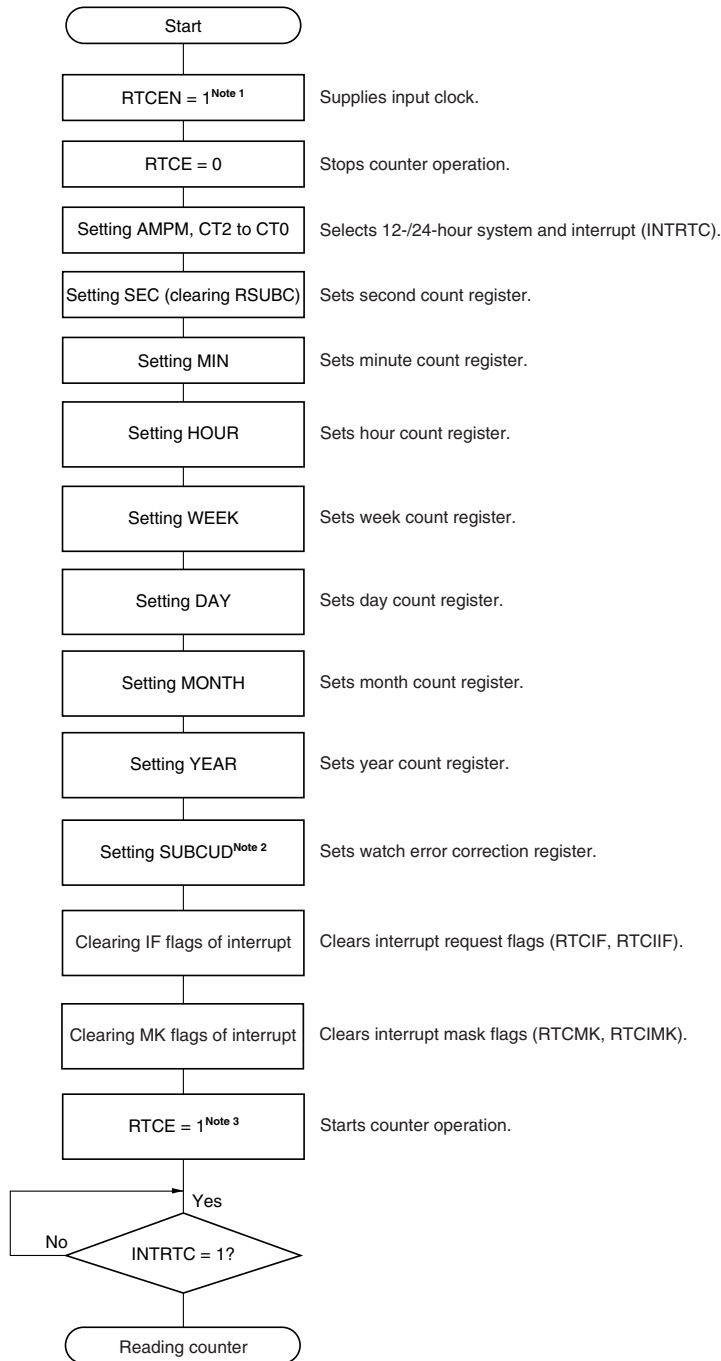
Symbol	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PMmn	Pmn pin I/O mode selection (m = 1 and 3 ; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

8.4 Real-Time Counter Operation

8.4.1 Starting operation of real-time counter

Figure 8-19. Procedure for Starting Operation of Real-Time Counter



<R>

- Notes**
1. First set RTCEN to 1, while oscillation of the subsystem clock ( $f_{SUB}$ ) is stable.
  2. Set up SUBCUD only if the watch error must be corrected. For details about how to calculate the correction value, see **8.4.8 Example of watch error correction of real-time counter**.
  3. Confirm the procedure described in **8.4.2 Shifting to STOP mode after starting operation** when shifting to STOP mode without waiting for INTRTC = 1 after RTCE = 1.

<R>

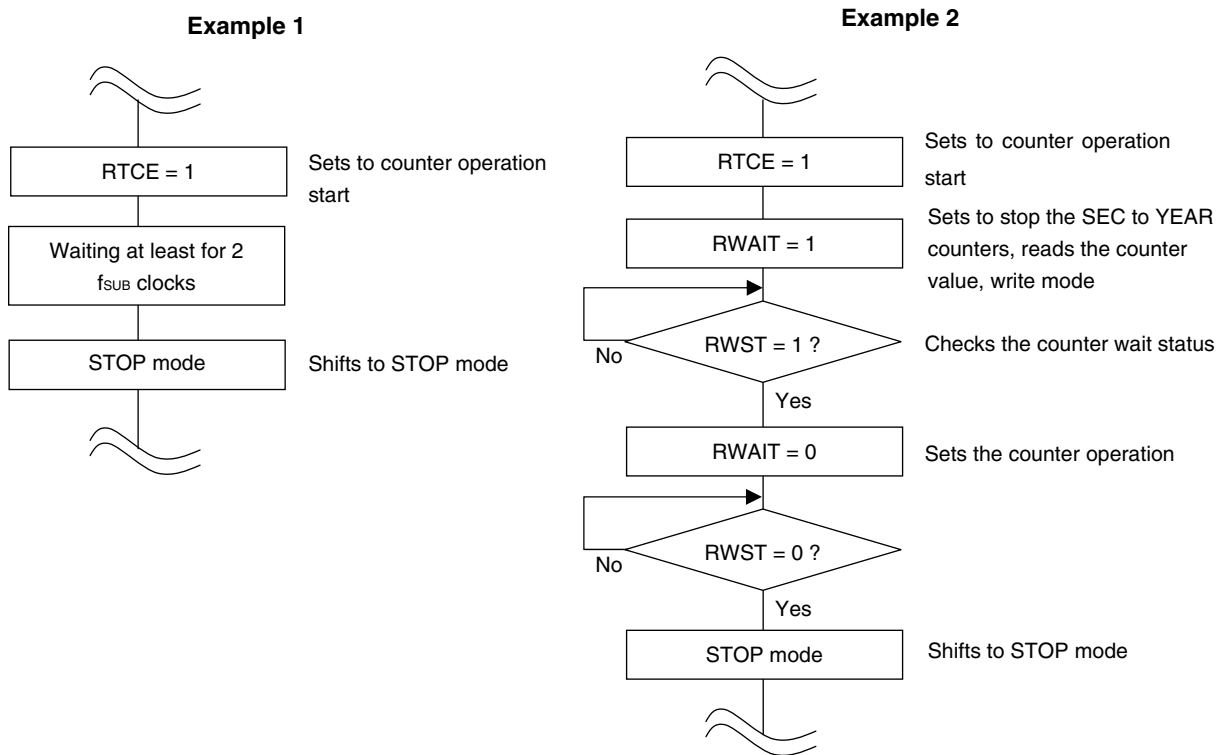
**8.4.2 Shifting to STOP mode after starting operation**

Perform one of the following processing when shifting to STOP mode immediately after setting RTCE to 1.

However, after setting RTCE to 1, this processing is not required when shifting to STOP mode after the first INTRTC interrupt has occurred.

- Shifting to STOP mode when at least two subsystem clocks ( $f_{SUB}$ ) (about  $62 \mu s$ ) have elapsed after setting RTCE to 1 (see **Figure 8-20, Example 1**).
- Checking by polling RWST to become 1, after setting RTCE to 1 and then setting RWAIT to 1. Afterward, setting RWAIT to 0 and shifting to STOP mode after checking again by polling that RWST has become 0 (see **Figure 8-20, Example 2**).

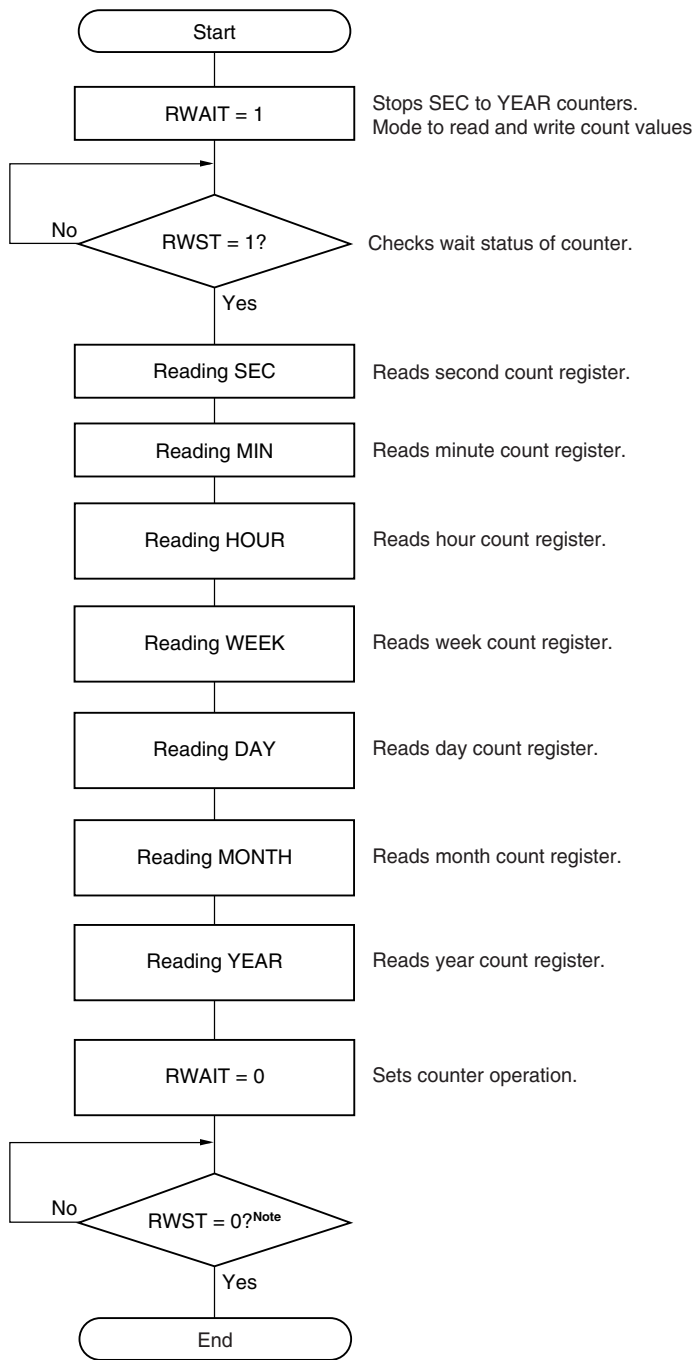
**Figure 8-20. Procedure for Shifting to STOP Mode After Setting RTCE to 1**



8.4.3 Reading/writing real-time counter

Read or write the counter after setting 1 to RWAIT first.

Figure 8-21. Procedure for Reading Real-Time Counter



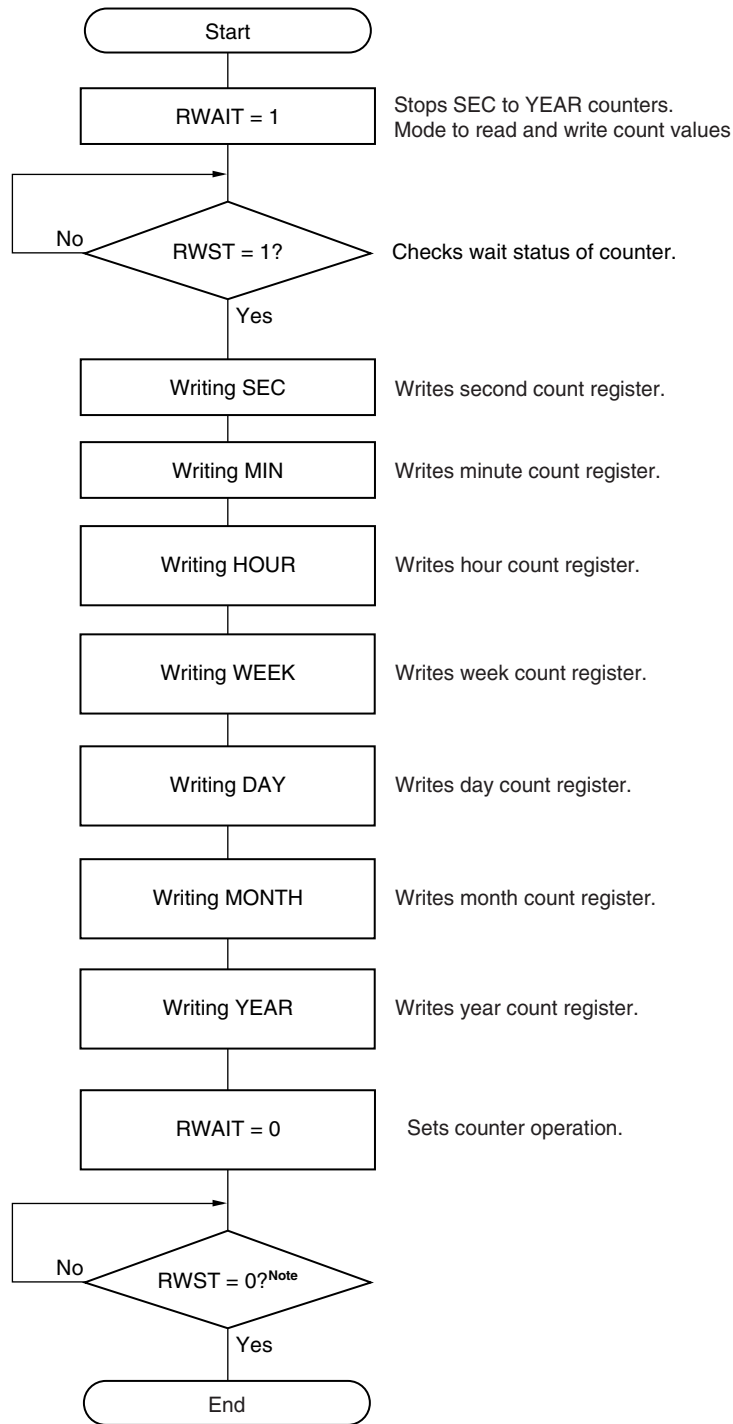
**Note** Be sure to confirm that RWST = 0 before setting STOP mode.

**Caution** Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

**Remark** SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence. All the registers do not have to be set and only some registers may be read.



Figure 8-22. Procedure for Writing Real-Time Counter



**Note** Be sure to confirm that RWST = 0 before setting STOP mode.

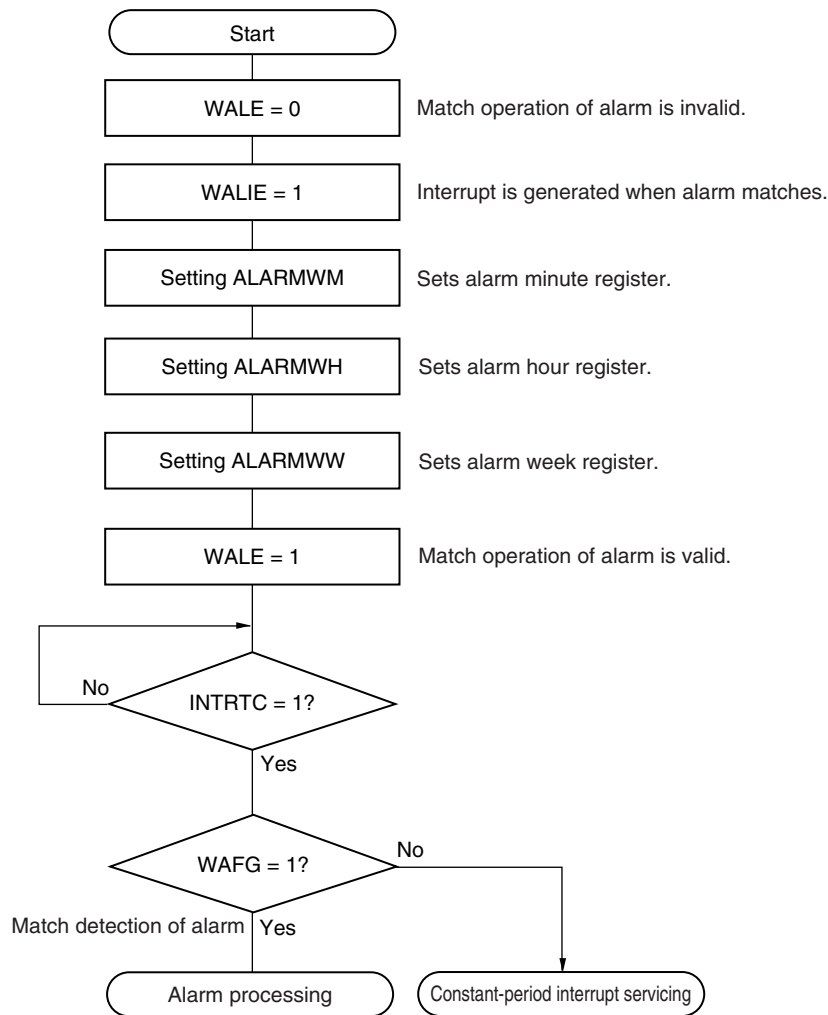
**Caution** Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

**Remark** SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be written in any sequence. All the registers do not have to be set and only some registers may be written.

#### 8.4.4 Setting alarm of real-time counter

Set time of alarm after setting 0 to WALE first.

Figure 8-23. Alarm Setting Procedure

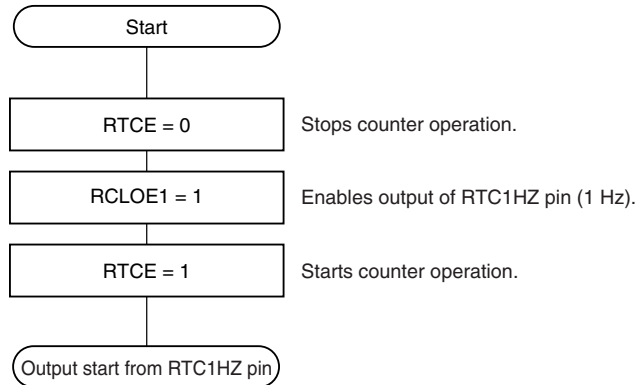


**Remarks 1.** ALARMWMM, ALARMWH, and ALARMWW may be written in any sequence.

- Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

8.4.5 1 Hz output of real-time counter

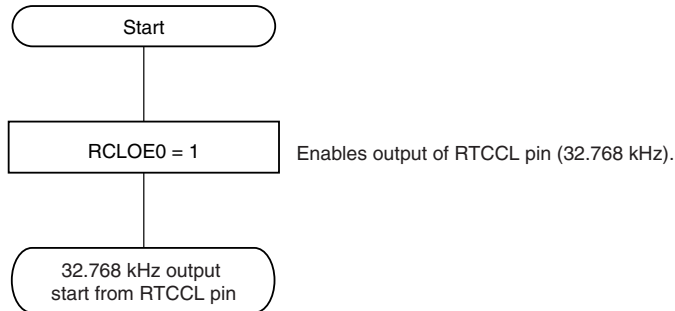
Figure 8-24. 1 Hz Output Setting Procedure



<R> **Caution** First set RTCEN to 1, while oscillation of the subsystem clock ( $f_{SUB}$ ) is stable.

<R> 8.4.6 32.768 kHz output of real-time counter

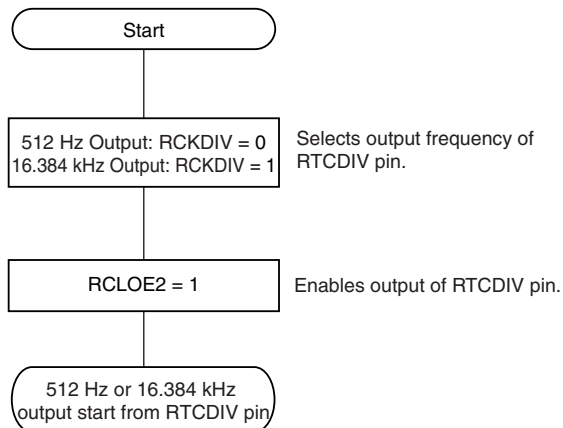
Figure 8-25. 32.768 kHz Output Setting Procedure



**Caution** First set RTCEN to 1, while oscillation of the subsystem clock ( $f_{SUB}$ ) is stable.

<R> 8.4.7 512 Hz, 16.384 kHz output of real-time counter

Figure 8-26. 512 Hz, 16.384 kHz output Setting Procedure



**Caution** First set RTCEN to 1, while oscillation of the subsystem clock ( $f_{SUB}$ ) is stable.

### 8.4.8 Example of watch error correction of real-time counter

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the watch error correction register.

#### Example of calculating the correction value

The correction value used when correcting the count value of the sub-count register (RSUBC) is calculated by using the following expression.

Set DEV to 0 when the correction range is  $-63.1$  ppm or less, or  $63.1$  ppm or more.

(When DEV = 0)

$$\text{Correction value}^{\text{Note}} = \text{Number of correction counts in 1 minute} \div 3 = (\text{Oscillation frequency} \div \text{Target frequency} - 1) \times 32768 \times 60 \div 3$$

(When DEV = 1)

$$\text{Correction value}^{\text{Note}} = \text{Number of correction counts in 1 minute} = (\text{Oscillation frequency} \div \text{Target frequency} - 1) \times 32768 \times 60$$

**Note** The correction value is the watch error correction value calculated by using bits 6 to 0 of the watch error correction register (SUBCUD).

$$\text{(When F6 = 0) Correction value} = \{(F5, F4, F3, F2, F1, F0) - 1\} \times 2$$

$$\text{(When F6 = 1) Correction value} = -\{(/F5, /F4, /F3, /F2, /F1, /F0) + 1\} \times 2$$

When (F6, F5, F4, F3, F2, F1, F0) is (\*, 0, 0, 0, 0, 0, \*), watch error correction is not performed. "\*" is 0 or 1.

/F5 to /F0 are bit-inverted values (000011 when 111100).

- Remarks**
1. The correction value is 2, 4, 6, 8, ... 120, 122, 124 or  $-2, -4, -6, -8, \dots -120, -122, -124$ .
  2. The oscillation frequency is the subsystem clock ( $f_{\text{SUB}}$ ).  
It can be calculated from the 32 kHz output frequency of the RTCCL pin or the output frequency of the RTC1HZ pin  $\times 32768$  when the watch error correction register is set to its initial value (00H).
  3. The target frequency is the frequency resulting after correction performed by using the watch error correction register.

**Correction example <1>**

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm)

[Measuring the oscillation frequency]

The oscillation frequency<sup>Note</sup> of each product is measured by outputting about 32 kHz from the RTCCL pin or outputting about 1 Hz from the RTC1HZ pin when the watch error correction register is set to its initial value (00H).

**Note** See **8.4.5 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **8.4.6 32.768 kHz output of real-time counter** for the setting procedure of outputting about 32 kHz from the RTCCL pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 32772.3 Hz)

If the target frequency is assumed to be 32768 Hz (32772.3 Hz – 131.2 ppm), the correction range for –131.2 ppm is –63.1 ppm or less, so assume DEV to be 0.

The expression for calculating the correction value when DEV is 0 is applied.

$$\begin{aligned}
 \text{Correction value} &= \text{Number of correction counts in 1 minute} \div 3 \\
 &= (\text{Oscillation frequency} \div \text{Target frequency} - 1) \times 32768 \times 60 \div 3 \\
 &= (32772.3 \div 32768 - 1) \times 32768 \times 60 \div 3 \\
 &= 86
 \end{aligned}$$

[Calculating the values to be set to (F6 to F0)]

(When the correction value is 86)

If the correction value is 0 or more (when delaying), assume F6 to be 0.

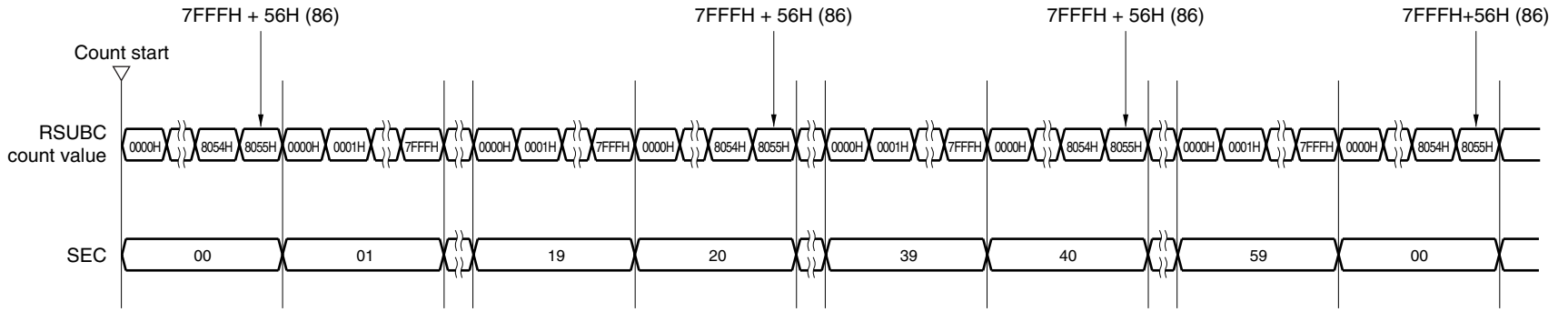
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

$$\begin{aligned}
 \{(F5, F4, F3, F2, F1, F0) - 1\} \times 2 &= 86 \\
 (F5, F4, F3, F2, F1, F0) &= 44 \\
 (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 0, 0)
 \end{aligned}$$

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm), setting the correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of SUBCUD: 0101100) results in 32768 Hz (0 ppm).

Figure 8-27 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 8-27. Operation When (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)



**Correction example <2>**

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

The oscillation frequency<sup>Note</sup> of each product is measured by outputting about 32 kHz from the RTCCL pin or outputting about 1 Hz from the RTC1HZ pin when the watch error correction register is set to its initial value (00H).

**Note** See **8.4.5 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **8.4.6 32.768 kHz output of real-time counter** for the setting procedure of outputting about 32 kHz from the RTCCL pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 0.9999817 Hz)

Oscillation frequency =  $32768 \times 0.9999817 \approx 32767.4$  Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

$$\begin{aligned} \text{Correction value} &= \text{Number of correction counts in 1 minute} \\ &= (\text{Oscillation frequency} \div \text{Target frequency} - 1) \times 32768 \times 60 \\ &= (32767.4 \div 32768 - 1) \times 32768 \times 60 \\ &= -36 \end{aligned}$$

[Calculating the values to be set to (F6 to F0)]

(When the correction value is -36)

If the correction value is 0 or less (when speeding up), assume F6 to be 1.

Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

$$\begin{aligned} -\{(/F5, /F4, /F3, /F2, /F1, /F0) + 1\} \times 2 &= -36 \\ (/F5, /F4, /F3, /F2, /F1, /F0) &= 17 \\ (/F5, /F4, /F3, /F2, /F1, /F0) &= (0, 1, 0, 0, 0, 1) \\ (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 1, 0) \end{aligned}$$

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is -36 (bits 6 to 0 of SUBCUD: 1101110) results in 32768 Hz (0 ppm).

Figure 8-28 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).





## CHAPTER 9 WATCHDOG TIMER

### 9.1 Functions of Watchdog Timer

The watchdog timer operates on the internal low-speed oscillation 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 WDTE
- If data is written to WDTE during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDRF) of the reset control flag register (RESF) is set to 1. For details of RESF, see **CHAPTER 20 RESET FUNCTION**.

When 75% of the overflow time is reached, an interval interrupt can be generated.

## 9.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

**Table 9-1. Configuration of Watchdog Timer**

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

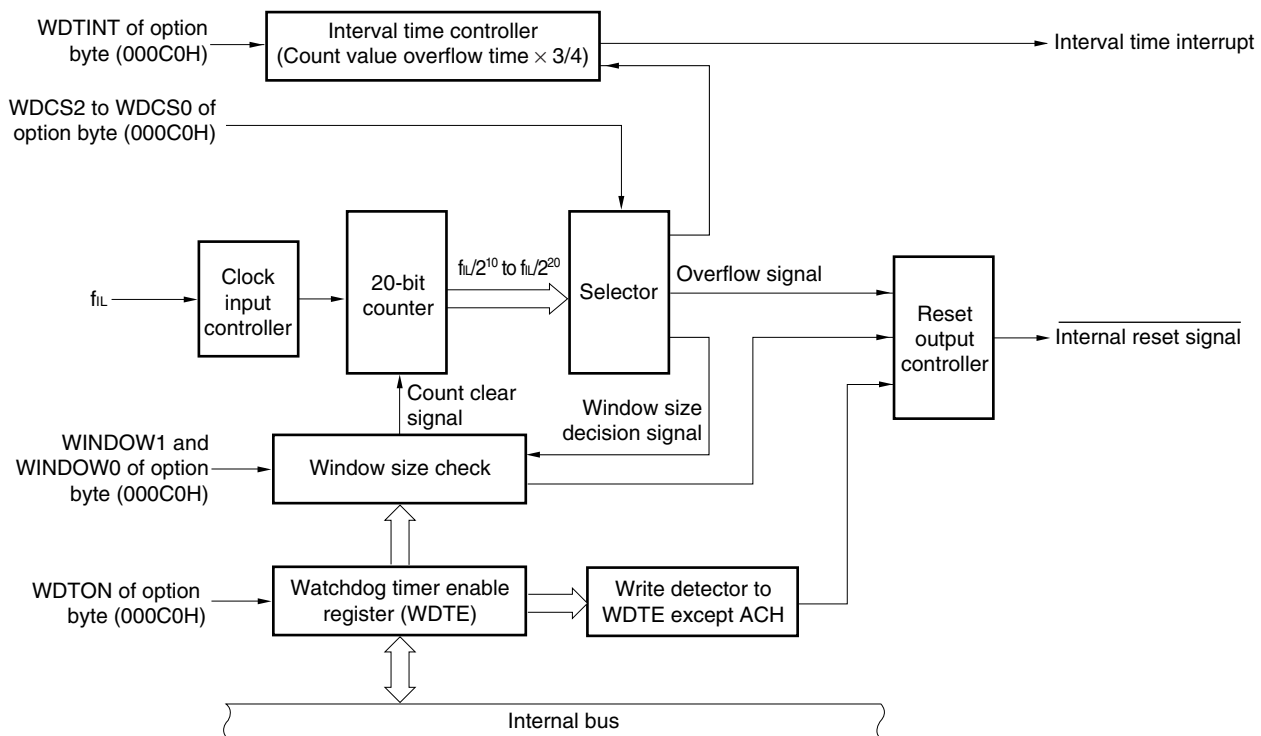
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

**Table 9-2. Setting of Option Bytes and Watchdog Timer**

Setting of Watchdog Timer	Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

**Remark** For the option byte, see **CHAPTER 24 OPTION BYTE**.

**Figure 9-1. Block Diagram of Watchdog Timer**



**Remark**  $f_{IL}$ : Internal low-speed oscillation clock frequency

### 9.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

#### (1) Watchdog timer enable register (WDTE)

Writing “ACH” to WDTE clears the watchdog timer counter and starts counting again.

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

Reset signal generation sets this register to 9AH or 1AH<sup>Note</sup>.

**Figure 9-2. Format of Watchdog Timer Enable Register (WDTE)**

Address: FFFABH	After reset: 9AH/1AH <sup>Note</sup>	R/W						
Symbol	7	6	5	4	3	2	1	0
WDTE								

**Note** The WDTE reset value differs depending on the WDTON setting value of the option byte (000C0H). To operate watchdog timer, set WDTON to 1.

WDTON Setting Value	WDTE 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 WDTE, an internal reset signal is generated.
  2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated.
  3. The value read from WDTE is 9AH/1AH (this differs from the written value (ACH)).

## 9.4 Operation of Watchdog Timer

### 9.4.1 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 24**).

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 **9.4.2** and **CHAPTER 24**).
  - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **9.4.3** and **CHAPTER 24**).
2. After a reset release, the watchdog timer starts counting.
  3. By writing “ACH” to WDTE after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
  4. After that, write WDTE the second time or later after a reset release during the window open period. If WDTE is written during a window close period, an internal reset signal is generated.
  5. If the overflow time expires without “ACH” written to WDTE, an internal reset signal is generated. A internal reset signal is generated in the following cases.
    - If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
    - If data other than “ACH” is written to WDTE

- Cautions**
1. When data is written to WDTE for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.
  2. If the watchdog timer is cleared by writing “ACH” to WDTE, the actual overflow time may be different from the overflow time set by the option byte by up to  $2/f_{IL}$  seconds.
  3. The watchdog timer can be cleared immediately before the count value overflows.

<Example> When the overflow time is set to  $2^{10}/f_{IL}$ , writing “ACH” is valid up to count value 3FH.

**Cautions 4.** 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	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		

If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts.

When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.

Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.

Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

5. The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

#### 9.4.2 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 WDTE during the window open period before the overflow time.

The following overflow time is set.

**Table 9-3. Setting of Overflow Time of Watchdog Timer**

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer
0	0	0	$2^{10}/f_{IL}$ (3.88 ms)
0	0	1	$2^{11}/f_{IL}$ (7.76 ms)
0	1	0	$2^{12}/f_{IL}$ (15.52 ms)
0	1	1	$2^{13}/f_{IL}$ (31.03 ms)
1	0	0	$2^{15}/f_{IL}$ (124.12 ms)
1	0	1	$2^{17}/f_{IL}$ (496.48 ms)
1	1	0	$2^{18}/f_{IL}$ (992.97 ms)
1	1	1	$2^{20}/f_{IL}$ (3971.88 ms)

**Caution** The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

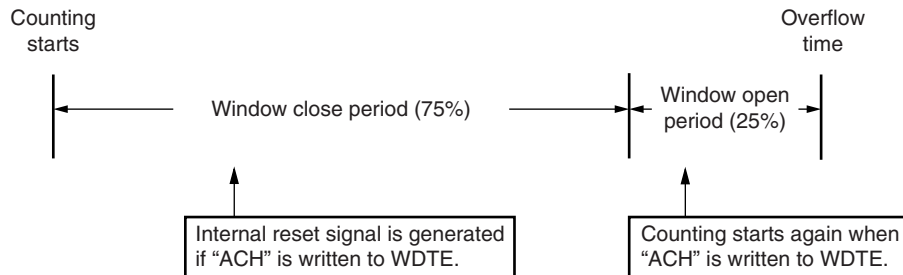
- Remarks**
1.  $f_{IL}$ : Internal low-speed oscillation clock frequency
  2. ( ):  $f_{IL} = 264$  kHz (MAX.)

### 9.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If “ACH” is written to WDTE during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to WDTE during the window close period, an abnormality is detected and an internal reset signal is generated.

**Example:** If the window open period is 25%



**Caution** When data is written to WDTE for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The window open period to be set is as follows.

**Table 9-4. Setting Window Open Period of Watchdog Timer**

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	25%
0	1	50%
1	0	75%
1	1	100%

- Cautions**
1. The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.
  2. When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of WINDOW1 and WINDOW0.
  3. Do not set the window open period to 25% if the watchdog timer corresponds to either of the conditions below.
    - When used at a supply voltage ( $V_{DD}$ ) below 2.7 V.
    - When stopping all main system clocks (internal high-speed oscillation clock, X1 clock, and external main system clock) by use of the STOP mode or software.
    - Low-power consumption mode

**Remarks 1.** If the overflow time is set to  $2^{10}/f_{IL}$ , the window close time and open time are as follows.

	Setting of Window Open Period			
	25%	50%	75%	100%
Window close time	0 to 3.56 ms	0 to 2.37 ms	0 to 0.119 ms	None
Window open time	3.56 to 3.88 ms	2.37 to 3.88 ms	0.119 to 3.88 ms	0 to 3.88 ms

<When window open period is 25%>

- Overflow time:  
 $2^{10}/f_{IL} \text{ (MAX.)} = 2^{10}/264 \text{ kHz (MAX.)} = 3.88 \text{ ms}$
- Window close time:  
 $0 \text{ to } 2^{10}/f_{IL} \text{ (MIN.)} \times (1 - 0.25) = 0 \text{ to } 2^{10}/216 \text{ kHz (MIN.)} \times 0.75 = 0 \text{ to } 3.56 \text{ ms}$
- Window open time:  
 $2^{10}/f_{IL} \text{ (MIN.)} \times (1 - 0.25) \text{ to } 2^{10}/f_{IL} \text{ (MAX.)} = 2^{10}/216 \text{ kHz (MIN.)} \times 0.75 \text{ to } 2^{10}/264 \text{ kHz (MAX.)}$   
 $= 3.56 \text{ to } 3.88 \text{ ms}$

2.  $f_{IL}$ : Internal low-speed oscillation clock frequency

#### 9.4.4 Setting watchdog timer interval interrupt

Depending on the setting of bit 7 (WDTINT) of an option byte (000C0H), an interval interrupt (INTWDTI) can be generated when 75% of the overflow time is reached.

**Table 9-5. Setting of Watchdog Timer Interval Interrupt**

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is used.
1	Interval interrupt is generated when 75% of overflow time is reached.

**Caution** When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed. Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset. Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

**Remark** The watchdog timer continues counting even after INTWDTI is generated (until ACH is written to the WDTE register). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

## CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

### 10.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral ICs.

Buzzer output is a function to output a square wave of buzzer frequency.

One pin can be used to output a clock or buzzer sound.

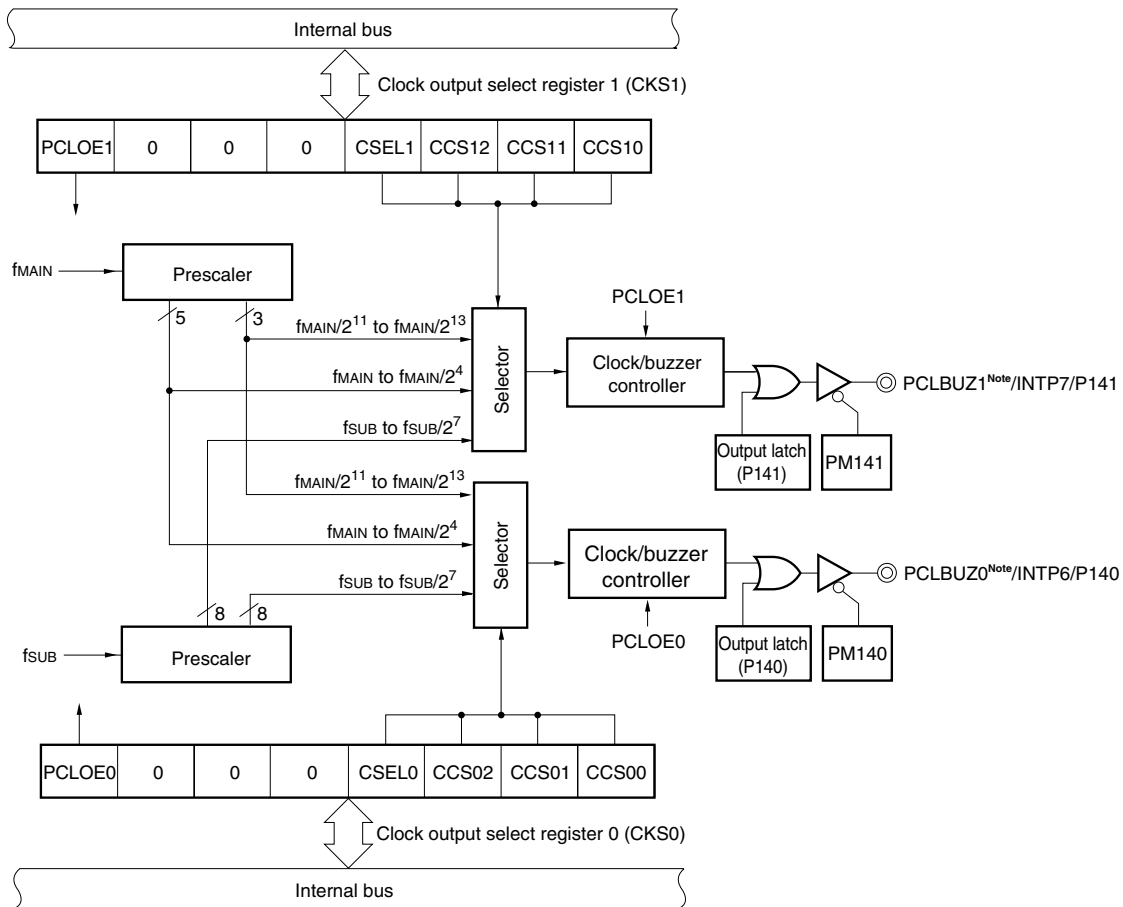
Two output pins, PCLBUZ0 and PCLBUZ1, are available.

PCLBUZ0 outputs a clock selected by clock output select register 0 (CKS0).

PCLBUZ1 outputs a clock selected by clock output select register 1 (CKS1).

Figure 10-1 shows the block diagram of clock output/buzzer output controller.

**Figure 10-1. Block Diagram of Clock Output/Buzzer Output Controller**



**Note** The PCLBUZ0 and PCLBUZ1 pins can output a clock of up to 10 MHz at  $2.7\text{ V} \leq V_{DD}$ . Setting a clock exceeding 5 MHz at  $V_{DD} < 2.7\text{ V}$  is prohibited.

**Remark**  $f_{MAIN}$ : Main system clock frequency  
 $f_{SUB}$ : Subsystem clock frequency



## 10.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

**Table 10-1. Configuration of Clock Output/Buzzer Output Controller**

Item	Configuration
Control registers	Clock output select registers 0, 1 (CKS0, CKS1) Port mode register 14 (PM14) Port register 14 (P14)

## 10.3 Registers Controlling Clock Output/Buzzer Output Controller

The following two registers are used to control the clock output/buzzer output controller.

- Clock output select registers 0, 1 (CKS0, CSK1)
- Port mode register 14 (PM14)

### (1) Clock output select registers 0, 1 (CKS0, CKS1)

These registers set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZ0/PCLBUZ1), and set the output clock.

Select the clock to be output from PCLBUZ0 by using CKS0.

Select the clock to be output from PCLBUZ1 by using CKS1.

CKS0 and CKS1 are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 10-2. Format of Clock Output Select Register n (CKSn)

Address: FFFA5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn output enable/disable specification
0	Output disable (default)
1	Output enable

CSELn	CCSn2	CCSn1	CCSn0		PCLBUZn output clock selection		
					f <sub>MAIN</sub> = 5 MHz	f <sub>MAIN</sub> = 10 MHz	f <sub>MAIN</sub> = 20 MHz
0	0	0	0	f <sub>MAIN</sub>	5 MHz	10 MHz <sup>Note</sup>	Setting prohibited <sup>Note</sup>
0	0	0	1	f <sub>MAIN</sub> /2	2.5 MHz	5 MHz	10 MHz <sup>Note</sup>
0	0	1	0	f <sub>MAIN</sub> /2 <sup>2</sup>	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	f <sub>MAIN</sub> /2 <sup>3</sup>	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	f <sub>MAIN</sub> /2 <sup>4</sup>	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	f <sub>MAIN</sub> /2 <sup>11</sup>	2.44 kHz	4.88 kHz	9.76 kHz
0	1	1	0	f <sub>MAIN</sub> /2 <sup>12</sup>	1.22 kHz	2.44 kHz	4.88 kHz
0	1	1	1	f <sub>MAIN</sub> /2 <sup>13</sup>	610 Hz	1.22 kHz	2.44 kHz
1	0	0	0	f <sub>SUB</sub>	32.768 kHz		
1	0	0	1	f <sub>SUB</sub> /2	16.384 kHz		
1	0	1	0	f <sub>SUB</sub> /2 <sup>2</sup>	8.192 kHz		
1	0	1	1	f <sub>SUB</sub> /2 <sup>3</sup>	4.096 kHz		
1	1	0	0	f <sub>SUB</sub> /2 <sup>4</sup>	2.048 kHz		
1	1	0	1	f <sub>SUB</sub> /2 <sup>5</sup>	1.024 kHz		
1	1	1	0	f <sub>SUB</sub> /2 <sup>6</sup>	512 Hz		
1	1	1	1	f <sub>SUB</sub> /2 <sup>7</sup>	256 Hz		

**Note** Setting an output clock exceeding 10 MHz is prohibited when  $2.7\text{ V} \leq V_{DD}$ . Setting a clock exceeding 5 MHz at  $V_{DD} < 2.7\text{ V}$  is also prohibited.

- Cautions**
1. Change the output clock after disabling clock output (PCLOEn = 0).
  2. If the selected clock (f<sub>MAIN</sub> or f<sub>SUB</sub>) stops during clock output (PCLOEn = 1), the output becomes undefined.

- Remarks**
1. n = 0, 1
  2. f<sub>MAIN</sub>: Main system clock frequency
  3. f<sub>SUB</sub>: Subsystem clock frequency

**(2) Port mode register 14 (PM14)**

This register sets port 14 input/output in 1-bit units.

When using the P140/INTP6/PCLBUZ0 and P141/INTP7/PCLBUZ1 pins for clock output/buzzer output, clear PM140 and PM141 and the output latches of P140 and P141 to 0.

PM14 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

**Figure 10-3. Format of Port Mode Register 14 (PM14)**

Address: FFF2EH    After reset: FFH    R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140

PM14n	P14n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

**10.4 Operations of Clock Output/Buzzer Output Controller**

One pin can be used to output a clock or buzzer sound.

Two output pins, PCLBUZ0 and PCLBUZ1, are available.

PCLBUZ0 outputs a clock/buzzer selected by clock output select register 0 (CKS0).

PCLBUZ1 outputs a clock/buzzer selected by clock output select register 1 (CKS1).

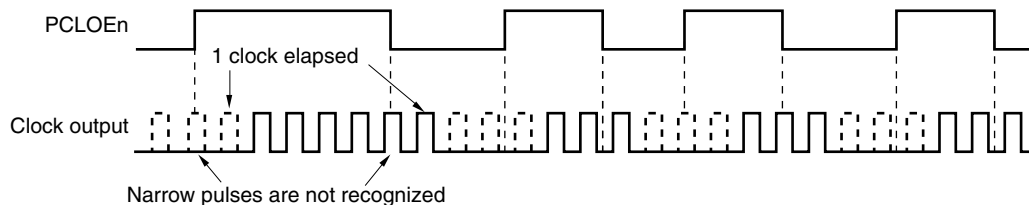
**10.4.1 Operation as output pin**

PCLBUZn is output as the following procedure.

- <1> Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of the clock output select register (CKSn) of the PCLBUZn pin (output in disabled status).
- <2> Set bit 7 (PCLOEn) of CKSn to 1 to enable clock/buzzer output.

- <R> **Remarks 1.** The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOEn) is switched. At this time, pulses with a narrow width are not output. Figure 10-4 shows enabling or stopping output using PCLOEn and the timing of outputting the clock.
2. n = 0, 1

**Figure 10-4. Remote Control Output Application Example**



## CHAPTER 11 A/D CONVERTER

### 11.1 Function of A/D Converter

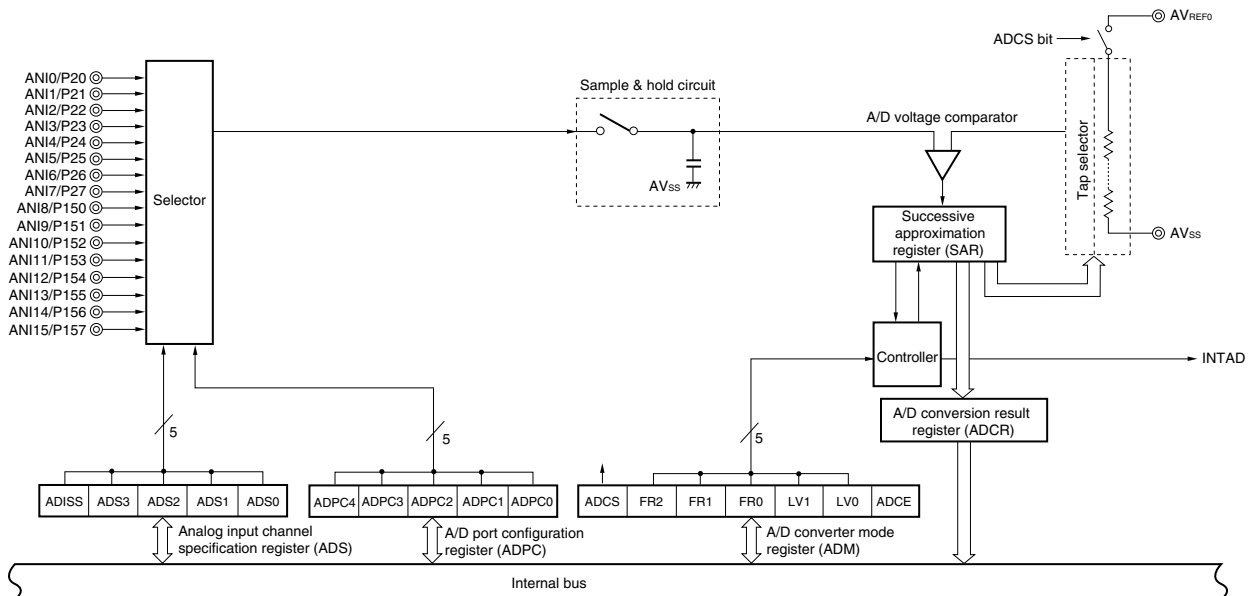
The A/D converter converts an analog input signal into a digital value, and consists of up to 16 channels (ANI0 to ANI15) with a resolution of 10 bits.

The A/D converter has the following function.

- **10-bit resolution A/D conversion**

10-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI15. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Figure 11-1. Block Diagram of A/D Converter



## 11.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

### (1) ANI0 to ANI15 pins

These are the analog input pins of the 16-channel 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.

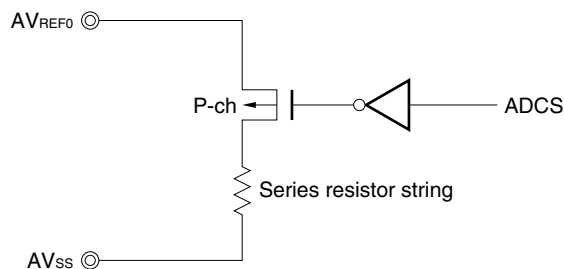
### (2) Sample & hold circuit

The sample & hold circuit samples the input voltage of the analog input pin selected by the selector when A/D conversion is started, and holds the sampled voltage value during A/D conversion.

### (3) Series resistor string

The series resistor string is connected between  $AV_{REF0}$  and  $AV_{SS}$ , and generates a voltage to be compared with the sampled voltage value.

Figure 11-2. Circuit Configuration of Series Resistor String



### (4) A/D Voltage comparator

The A/D voltage comparator compares the sampled voltage value and the output voltage of the series resistor string.

### (5) Successive approximation register (SAR)

This register converts the result of comparison by the A/D voltage comparator, starting from the most significant bit (MSB).

When the voltage value is converted into a digital value down to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred to the A/D conversion result register (ADCR).

### (6) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

### (7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

### (8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD.

### (9) AV<sub>REF0</sub> pin

This pin inputs an analog power/reference voltage to the A/D converter. The signal input to ANI0 to ANI15 is converted into a digital signal, based on the voltage applied across AV<sub>REF0</sub> and AV<sub>SS</sub>. The voltage that can be supplied to AV<sub>REF0</sub> varies as follows, depending on whether P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 are used as digital I/Os or analog inputs.

**Table 11-1. AV<sub>REF0</sub> Voltage Applied to P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 Pins**

Analog/Digital	V <sub>DD</sub> Condition	AV <sub>REF0</sub> Voltage
Using at least one pin as an analog input and using all pins not as digital I/Os	$2.3\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$2.3\text{ V} \leq AV_{REF0} \leq V_{DD} = EV_{DD0} = EV_{DD1}$
Pins used as analog inputs and digital I/Os are mixed <sup>Note</sup>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$2.7\text{ V} \leq AV_{REF0} \leq V_{DD} = EV_{DD0} = EV_{DD1}$
	$2.3\text{ V} \leq V_{DD} < 2.7\text{ V}$	AV <sub>REF0</sub> has same potential as EV <sub>DD0</sub> , EV <sub>DD1</sub> , and V <sub>DD</sub>
Using at least one pin as a digital I/O and using all pins not as analog inputs <sup>Note</sup>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$2.7\text{ V} \leq AV_{REF0} \leq V_{DD} = EV_{DD0} = EV_{DD1}$
	$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	AV <sub>REF0</sub> has same potential as EV <sub>DD0</sub> , EV <sub>DD1</sub> , and V <sub>DD</sub>

**Note** AV<sub>REF0</sub> is the reference for the I/O voltage of a port to be used as a digital port.

- High-/low-level input voltage (V<sub>IH4</sub>/V<sub>IL4</sub>)
- High-/low-level output voltage (V<sub>OH2</sub>/V<sub>OL2</sub>)

### (10) AV<sub>SS</sub> pin

This is the ground potential pin of the A/D converter. Always use this pin at the same potential as that of the EV<sub>SS0</sub>, EV<sub>SS1</sub>, and V<sub>SS</sub> pins even when the A/D converter is not used.

### (11) A/D converter mode register (ADM)

This register is used to set the conversion time of the analog input signal to be converted, and to start or stop the conversion operation.

### (12) A/D port configuration register (ADPC)

This register switches the ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157 pins to analog input of A/D converter or digital I/O of port.

### (13) Analog input channel specification register (ADS)

This register is used to specify the port that inputs the analog voltage to be converted into a digital signal.

### (14) Port mode registers 2 and 15 (PM2, PM15)

This register switches the ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157 pins to input or output.

### 11.3 Registers Used in A/D Converter

The A/D converter uses the following seven registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register (ADM)
- A/D port configuration register (ADPC)
- Analog input channel specification register (ADS)
- Port mode registers 2 and 15 (PM2, PM15)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)

#### (1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the A/D converter is used, be sure to set bit 5 (ADCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

**Figure 11-3. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU0EN	TAU0EN
ADCEN	Control of A/D converter input clock							
0	Stops supply of input clock. <ul style="list-style-type: none"> <li>• SFR used by the A/D converter cannot be written.</li> <li>• The A/D converter is in the reset status.</li> </ul>							
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the A/D converter can be read/written.</li> </ul>							

**Caution** When setting the A/D converter, be sure to set ADCEN 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 port mode registers 2 and 15 (PM2, PM15)).

**(2) A/D converter mode register (ADM)**

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion. ADM can be set by a 1-bit or 8-bit memory manipulation instruction. Reset signal generation clears this register to 00H.

**Figure 11-4. Format of A/D Converter Mode Register (ADM)**

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	FR2 <sup>Note 1</sup>	FR1 <sup>Note 1</sup>	FR0 <sup>Note 1</sup>	LV1 <sup>Note 1</sup>	LV0 <sup>Note 1</sup>	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation
1	Enables conversion operation

ADCE	A/D voltage comparator operation control <sup>Note 2</sup>
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

- Notes**
1. For details of FR2 to FR0, LV1, LV0, and A/D conversion, see **Table 11-3 A/D Conversion Time Selection**.
  2. The operation of the A/D voltage comparator is controlled by ADCS and ADCE, and it takes 1  $\mu$ s from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1  $\mu$ s or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

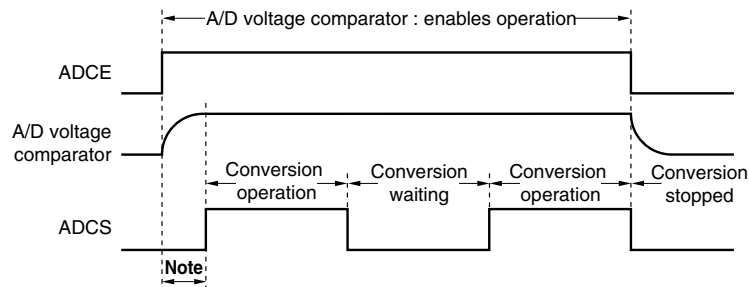
<R>

**Table 11-2. Settings of ADCS and ADCE**

ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion waiting mode (only A/D voltage comparator consumes power)
1	0	Setting prohibited
1	1	Conversion mode (A/D voltage comparator: enables operation)

<R>

**Figure 11-5. Timing Chart When A/D voltage Comparator Is Used**



**Note** To stabilize the internal circuit, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1  $\mu$ s or longer.

**Caution** A/D conversion must be stopped before rewriting bits FR0 to FR2, LV1, and LV0 to values other than the identical data.



Table 11-3. A/D Conversion Time Selection

(1)  $2.7\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ 

A/D Converter Mode Register (ADM)					Conversion Time Selection			Conversion Clock ( $f_{AD}$ )	
FR2	FR1	FR0	LV1	LV0	$f_{CLK} = 2\text{ MHz}$	$f_{CLK} = 10\text{ MHz}$	$f_{CLK} = 20\text{ MHz}$		
0	0	0	0	0	$264/f_{CLK}$	Setting prohibited	$26.4\ \mu\text{s}$	$13.2\ \mu\text{s}$	$f_{CLK}/12$
0	0	1	0	0	$176/f_{CLK}$		$17.6\ \mu\text{s}$	$8.8\ \mu\text{s}^{\text{Note 1}}$	$f_{CLK}/8$
0	1	0	0	0	$132/f_{CLK}$	$66.0\ \mu\text{s}^{\text{Note 2}}$	$13.2\ \mu\text{s}$	$6.6\ \mu\text{s}^{\text{Note 1}}$	$f_{CLK}/6$
0	1	1	0	0	$88/f_{CLK}$	$44.0\ \mu\text{s}^{\text{Note 2}}$	$8.8\ \mu\text{s}^{\text{Note 1}}$	Setting prohibited	$f_{CLK}/4$
1	0	0	0	0	$66/f_{CLK}$	$33.0\ \mu\text{s}$	$6.6\ \mu\text{s}^{\text{Note 1}}$		$f_{CLK}/3$
1	0	1	0	0	$44/f_{CLK}$	$22.0\ \mu\text{s}$	Setting prohibited		$f_{CLK}/2$
1	1	1	0	0	$22/f_{CLK}$	$11.0\ \mu\text{s}^{\text{Note 1}}$			$f_{CLK}$
Other than above					Setting prohibited				

- Notes**
- This can be set only when  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ .
  - Functionally expanded products ( $\mu\text{PD78F117xA}$ ) only.

**Caution** Set the conversion times with the following conditions.

**Conventional-specification products ( $\mu\text{PD78F117x}$ )**

- $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ :  $f_{AD} = 0.6$  to  $3.6\text{ MHz}$
- $2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$ :  $f_{AD} = 0.6$  to  $1.8\text{ MHz}$

**Functionally expanded products ( $\mu\text{PD78F117xA}$ )**

- $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ :  $f_{AD} = 0.33$  to  $3.6\text{ MHz}$
- $2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$ :  $f_{AD} = 0.33$  to  $1.8\text{ MHz}$

(2)  $2.3\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ 

A/D Converter Mode Register (ADM)					Conversion Time Selection			Conversion Clock ( $f_{AD}$ )
FR2	FR1	FR0	LV1	LV0		$f_{CLK} = 2\text{ MHz}$	$f_{CLK} = 5\text{ MHz}$	
0	0	0	0	1	$480/f_{CLK}$	Setting prohibited	Setting prohibited	$f_{CLK}/12$
0	0	1	0	1	$320/f_{CLK}$		$64.0\ \mu\text{s}$	$f_{CLK}/8$
0	1	0	0	1	$240/f_{CLK}$		$48.0\ \mu\text{s}$	$f_{CLK}/6$
0	1	1	0	1	$160/f_{CLK}$		$32.0\ \mu\text{s}$	$f_{CLK}/4$
1	0	0	0	1	$120/f_{CLK}$	$60.0\ \mu\text{s}$	$24.0\ \mu\text{s}^{\text{Note 1}}$	$f_{CLK}/3$
1	0	1	0	1	$80/f_{CLK}$	$40.0\ \mu\text{s}$	$16.0\ \mu\text{s}^{\text{Note 2}}$	$f_{CLK}/2$
1	1	1	0	1	$40/f_{CLK}$	$20.0\ \mu\text{s}^{\text{Note 2}}$	Setting prohibited	$f_{CLK}$
Other than above					Setting prohibited			

**Notes 1.** This can be set only when  $2.7\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ .

**2.** This can be set only when  $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ .

**Cautions 1. Set the conversion times with the following conditions.**

- $4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ :  $f_{AD} = 0.6\text{ to }3.6\text{ MHz}$
- $2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$ :  $f_{AD} = 0.6\text{ to }1.8\text{ MHz}$
- $2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$ :  $f_{AD} = 0.6\text{ to }1.44\text{ MHz}$

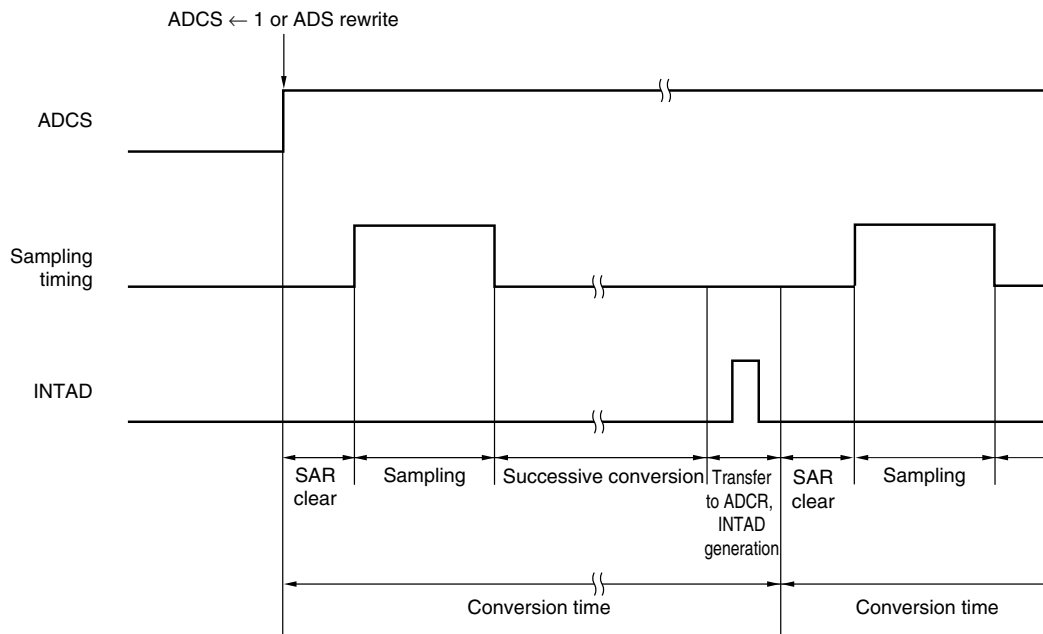
**2. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.**

**3. Change LV1 and LV0 from the default value, when  $2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$ .**

**4. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.**

**Remark**  $f_{CLK}$ : CPU/peripheral hardware clock frequency

Figure 11-6. A/D Converter Sampling and A/D Conversion Timing

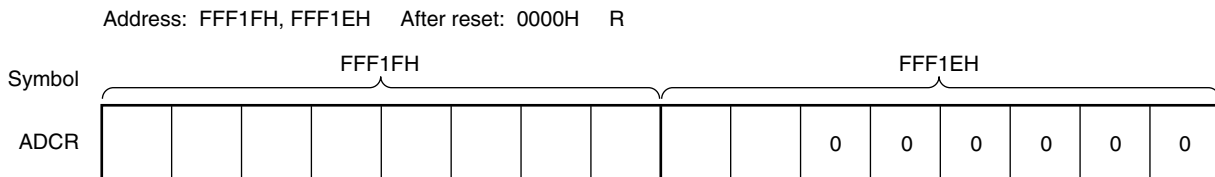


**(3) 10-bit A/D conversion result register (ADCR)**

This register is a 16-bit register that stores the A/D conversion result. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register. The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH. ADCR can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

**Figure 11-7. Format of 10-Bit A/D Conversion Result Register (ADCR)**



**Caution** When writing to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.

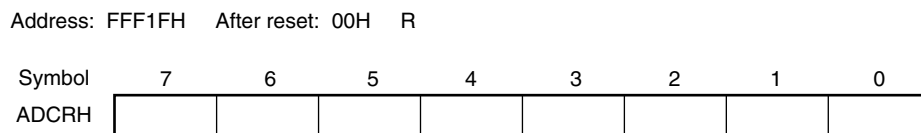
**(4) 8-bit A/D conversion result register (ADCRH)**

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored.

ADCRH can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 11-8. Format of 8-Bit A/D Conversion Result Register (ADCRH)**



**Caution** When writing to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.

**(5) Analog input channel specification register (ADS)**

This register specifies the input channel of the analog voltage to be A/D converted.

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

Reset signal generation clears this register to 00H.

**Figure 11-9. Format of Analog Input Channel Specification Register (ADS)**

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	0	ADS3	ADS2	ADS1	ADS0

ADISS	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	ANI0	P20/ANI0 pin
0	0	0	0	1	ANI1	P21/ANI1 pin
×	0	0	1	0	ANI2	P22/ANI2 pin
×	0	0	1	1	ANI3	P23/ANI3 pin
×	0	1	0	0	ANI4	P24/ANI4 pin
×	0	1	0	1	ANI5	P25/ANI5 pin
×	0	1	1	0	ANI6	P26/ANI6 pin
×	0	1	1	1	ANI7	P27/ANI7 pin
×	1	0	0	0	ANI8	P150/ANI8 pin
×	1	0	0	1	ANI9	P151/ANI9 pin
×	1	0	1	0	ANI10	P152/ANI10 pin
×	1	0	1	1	ANI11	P153/ANI11 pin
×	1	1	0	0	ANI12	P154/ANI12 pin
×	1	1	0	1	ANI13	P155/ANI13 pin
×	1	1	1	0	ANI14	P156/ANI14 pin
×	1	1	1	1	ANI15	P157/ANI15 pin

- Cautions**
1. Be sure to clear bits 4 to 6 to “0”.
  2. Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 15 (PM2, PM15).
  3. Do not set the pin that is set by ADPC as digital I/O by ADS.

**Remark** ×: don't care

**(6) A/D port configuration register (ADPC)**

This register switches the ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157 pins to analog input of A/D converter or digital I/O of port.

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

Reset signal generation sets this register to 10H.

**Figure 11-10. Format of A/D Port Configuration Register (ADPC)**

Address: F0017H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	Analog Input (A)/digital I/O (D) switching															
					Port 15								Port 2							
					ANI15 /P157	ANI14 /P156	ANI13 /P155	ANI12 /P154	ANI11 /P153	ANI10 /P152	ANI9 /P151	ANI8 /P150	ANI7 /P27	ANI6 /P26	ANI5 /P25	ANI4 /P24	ANI3 /P23	ANI2 /P22	ANI1 /P21	ANI0 /P20
0	0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
0	0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	A	A	A	D	
0	0	0	1	0	A	A	A	A	A	A	A	A	A	A	A	A	A	D	D	
0	0	0	1	1	A	A	A	A	A	A	A	A	A	A	A	A	D	D	D	
0	0	1	0	0	A	A	A	A	A	A	A	A	A	A	A	D	D	D	D	
0	0	1	0	1	A	A	A	A	A	A	A	A	A	A	D	D	D	D	D	
0	0	1	1	0	A	A	A	A	A	A	A	A	D	D	D	D	D	D	D	
0	0	1	1	1	A	A	A	A	A	A	A	A	D	D	D	D	D	D	D	
0	1	0	0	0	A	A	A	A	A	A	A	D	D	D	D	D	D	D	D	
0	1	0	0	1	A	A	A	A	A	A	A	D	D	D	D	D	D	D	D	
0	1	0	1	0	A	A	A	A	A	A	D	D	D	D	D	D	D	D	D	
0	1	0	1	1	A	A	A	A	A	D	D	D	D	D	D	D	D	D	D	
0	1	1	0	0	A	A	A	A	D	D	D	D	D	D	D	D	D	D	D	
0	1	1	0	1	A	A	A	D	D	D	D	D	D	D	D	D	D	D	D	
0	1	1	1	0	A	A	D	D	D	D	D	D	D	D	D	D	D	D	D	
0	1	1	1	1	A	D	D	D	D	D	D	D	D	D	D	D	D	D	D	
1	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	
Other than above					Setting prohibited															

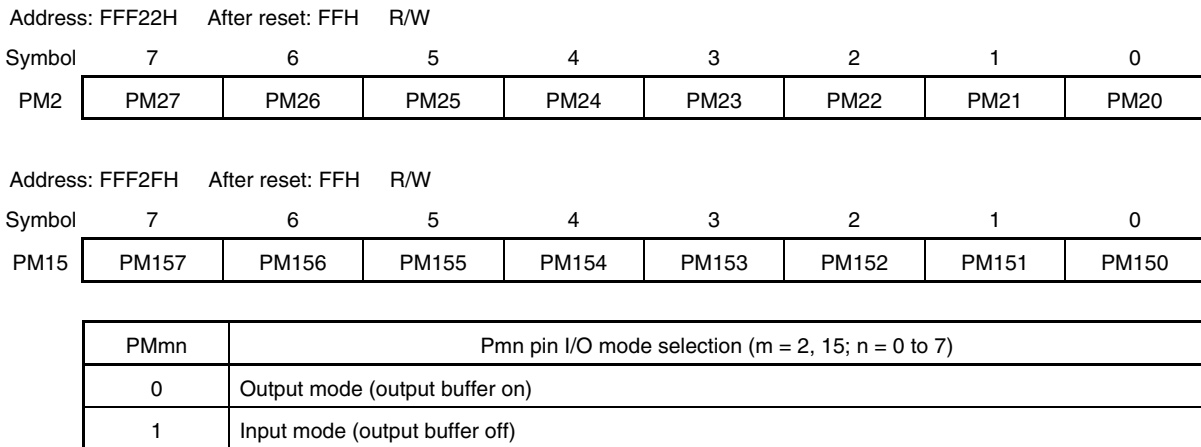
- Cautions**
1. Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 15 (PM2, PM15).
  2. Do not set the pin that is set by ADPC as digital I/O by ADS.
  3. P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 are set as analog inputs in the order of P157/ANI15, ..., P150/ANI8, P27/ANI7, ..., P20/ANI0 by the A/D port configuration register (ADPC). When using P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 as analog inputs, start designing from P157/ANI15.

**(7) Port mode registers 2 and 15 (PM2, PM15)**

When using the ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157 pins for analog input port, set PM20 to PM27 and PM150 to PM157 to 1. The output latches of P20 to P27 and P150 to P157 at this time may be 0 or 1. If PM20 to PM27 and PM150 to PM157 are set to 0, they cannot be used as analog input port pins. PM2 and PM15 can be set by a 1-bit or 8-bit memory manipulation instruction. Reset signal generation sets these registers to FFH.

**Caution** If a pin is set as an analog input port, not the pin level but “0” is always read.

**Figure 11-11. Format of Port Mode Registers 2 and 15 (PM2, PM15)**



ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157 pins are as shown below depending on the settings of ADPC, ADS, PM2, and PM15.

**Table 11-4. Setting Functions of ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157 Pins**

ADPC	PM2 and PM15	ADS	ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

## 11.4 A/D Converter Operations

### 11.4.1 Basic operations of A/D converter

- <1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1 to start the supply of the input clock to the A/D converter.
- <2> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1 to start the operation of the A/D voltage comparator.
- <3> Set channels for A/D conversion to analog input by using the A/D port configuration register (ADPC) and set to input mode by using port mode registers 2 and 15 (PM2, PM15).
- <4> Set A/D conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM.
- <5> Select one channel for A/D conversion using the analog input channel specification register (ADS).
- <6> Start the conversion operation by setting bit 7 (ADCS) of ADM to 1.  
(<7> to <13> are operations performed by hardware.)
- <7> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.
- <8> 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.
- <9> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to  $(1/2) AV_{REF0}$  by the tap selector.
- <10> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the A/D voltage comparator. If the analog input is greater than  $(1/2) AV_{REF0}$ , the MSB of SAR remains set to 1. If the analog input is smaller than  $(1/2) AV_{REF0}$ , the MSB is reset to 0.
- <11> Next, bit 8 of SAR is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.
  - Bit 9 = 1:  $(3/4) AV_{REF0}$
  - Bit 9 = 0:  $(1/4) AV_{REF0}$
 The voltage tap and sampled voltage are compared and bit 8 of SAR is manipulated as follows.
  - Analog input voltage  $\geq$  Voltage tap: Bit 8 = 1
  - Analog input voltage  $<$  Voltage tap: Bit 8 = 0
- <12> Comparison is continued in this way up to bit 0 of SAR.
- <13> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.  
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.
- <14> Repeat steps <7> to <13>, until ADCS is cleared to 0.  
To stop the A/D converter, clear ADCS to 0.  
To restart A/D conversion from the status of ADCE = 1, start from <6>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1  $\mu$ s or longer, and start <6>. To change a channel of A/D conversion, start from <5>.

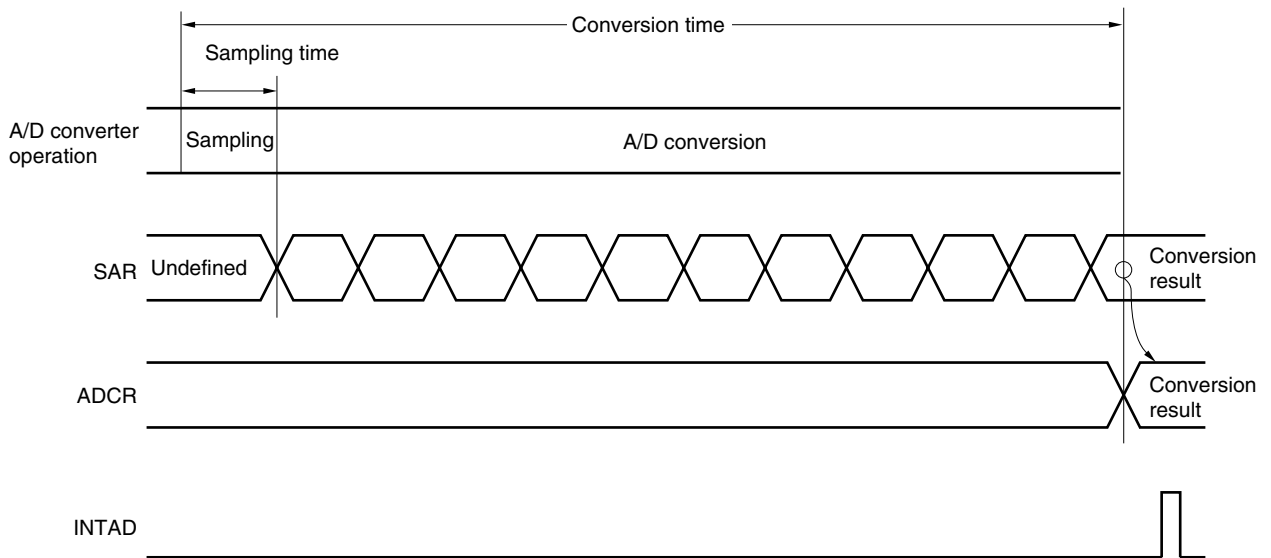
**Caution** Make sure the period of <2> to <6> is 1  $\mu$ s or more.

**Remark** Two types of A/D conversion result registers are available.

- ADCR (16 bits): Store 10-bit A/D conversion value
- ADCRH (8 bits): Store 8-bit A/D conversion value



Figure 11-12. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register (ADM) is reset (0) by software.

If a write operation is performed to the analog input channel specification register (ADS) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (1), conversion starts again from the beginning.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

**11.4.2 Input voltage and conversion results**

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI15) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT \left( \frac{V_{AIN}}{AV_{REF0}} \times 1024 + 0.5 \right)$$

$$ADCR = SAR \times 64$$

or

$$\left( \frac{ADCR}{64} - 0.5 \right) \times \frac{AV_{REF0}}{1024} \leq V_{AIN} < \left( \frac{ADCR}{64} + 0.5 \right) \times \frac{AV_{REF0}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V<sub>AIN</sub>: Analog input voltage

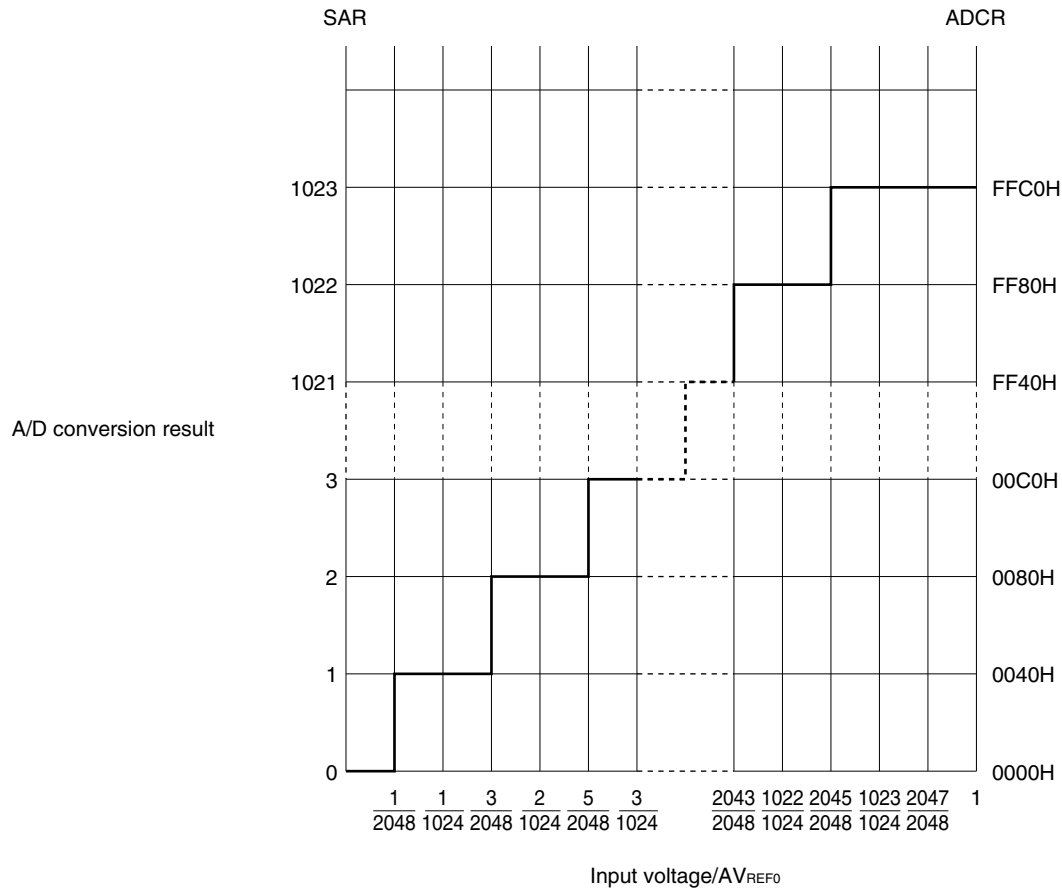
AV<sub>REF0</sub>: AV<sub>REF0</sub> pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 11-13 shows the relationship between the analog input voltage and the A/D conversion result.

**Figure 11-13. Relationship Between Analog Input Voltage and A/D Conversion Result**



### 11.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One channel of analog input is selected from ANI0 to ANI15 by the analog input channel specification register (ADS) and A/D conversion is executed.

#### (1) A/D conversion operation

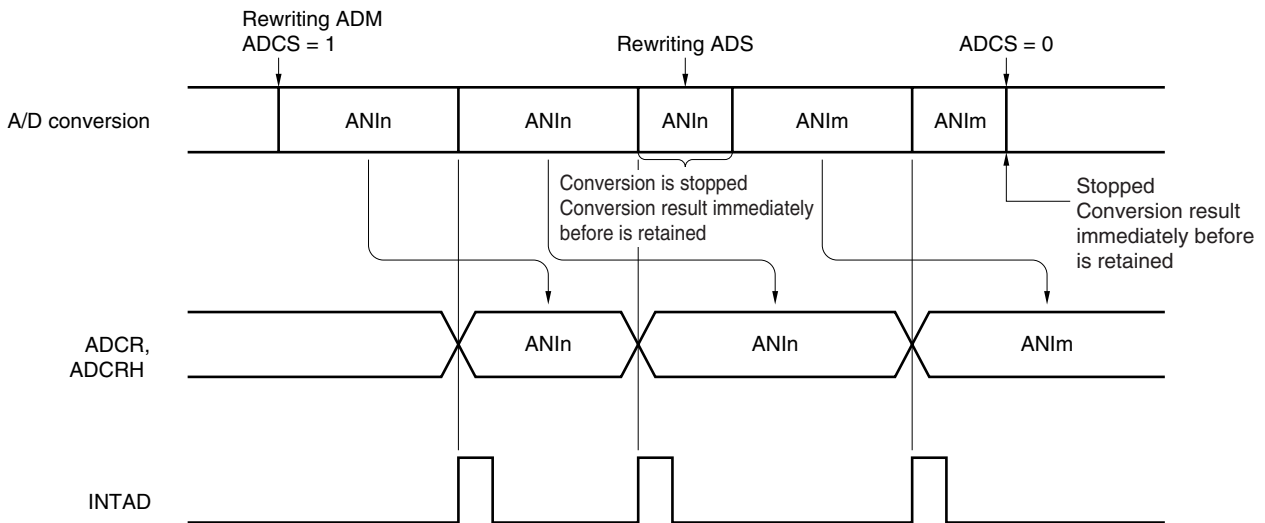
By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, the next A/D conversion operation is immediately started.

If ADS is rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning.

If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped. At this time, the conversion result immediately before is retained.

Figure 11-14. A/D Conversion Operation



- Remarks 1.** n = 0 to 15  
**2.** m = 0 to 15

The setting methods are described below.

- <1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1.
  - <2> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.
  - <3> Set the channel to be used in the analog input mode by using bits 4 to 0 (ADPC4 to ADPC0) of the A/D port configuration register (ADPC) and bits 7 to 0 (PM27 to PM20) of port mode register 2 (PM2) and bits 7 to 0 (PM157 to PM150) of port mode register 15 (PM15).
  - <4> Select conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM.
  - <5> Select a channel to be used by using bits 7 and 3 to 0 (ADISS, ADS3 to ADS0) of the analog input channel specification register (ADS).
  - <6> Set bit 7 (ADCS) of ADM to 1 to start A/D conversion.
  - <7> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
  - <8> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).
- <Change the channel>
- <9> Change the channel using bits 7 and 3 to 0 (ADISS, ADS3 to ADS0) of ADS to start A/D conversion.
  - <10> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
  - <11> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).
- <Complete A/D conversion>
- <12> Clear ADCS to 0.
  - <13> Clear ADCE to 0.
  - <14> Clear bit 5 (ADCEN) of peripheral enable register 0 (PER0)

- Cautions**
1. Make sure the period of <2> to <6> is 1  $\mu$ s or more.
  2. <2> may be done between <3> and <5>.
  3. The period from <7> to <10> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM. The period from <9> to <10> is the conversion time set using FR2 to FR0, LV1, and LV0.

### 11.5 Temperature Sensor Function (Expanded-Specification Products ( $\mu$ PD78F117xA) Only)

A temperature sensor performs A/D conversion for two voltages, an internal reference voltage (sensor 0 on the ANI0 side) that depends on the temperature and an internal reference voltage (sensor 1 on the ANI1 side) that does not depend on the temperature, and calculations, so that the temperature is obtained without depending on the AVREF0 voltage ( $AV_{REF0} \geq 2.7$  V).

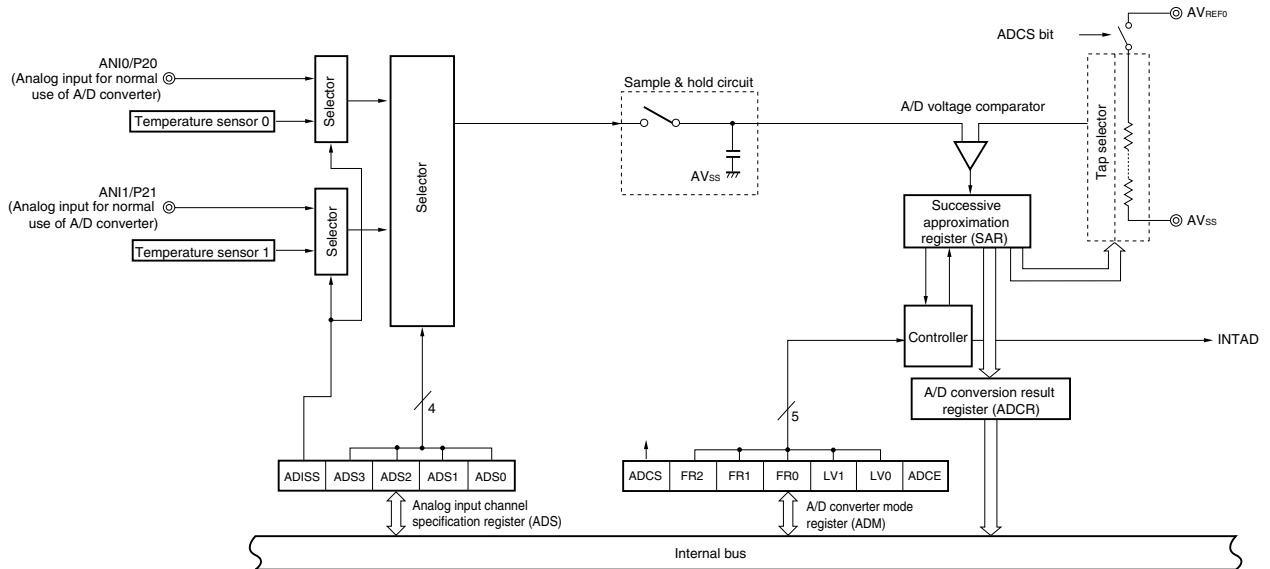
**Caution** The temperature sensor cannot be used when low current consumption mode is set ( $RMC = 5AH$ ) or when the internal high-speed oscillator has been stopped ( $HIOSTOP = 1$  (bit 0 of CSC register)). The temperature sensor can operate as long as the internal high-speed oscillator operates ( $HIOSTOP = 0$ ), even if it is not selected as the CPU/peripheral hardware clock source.

#### 11.5.1 Configuration of temperature sensor

The temperature sensor consists of an A/D converter and the following hardware.

- Temperature sensor 0: Outputs the internal reference voltage that depends on the temperature
- Temperature sensor 1: Outputs the internal reference voltage that does not depend on the temperature

Figure 11-15. Temperature Sensor Block Diagram



### 11.5.2 Registers used by temperature sensors

The following four types of registers are used when using a temperature sensor.

- Peripheral enable register 0 (PER0)
- A/D converter mode register (ADM)
- Analog input channel specification register (ADS)
- 10-bit A/D conversion result register (ADCR)

**Caution** Setting of the A/D port configuration register (ADPC), port mode register 2 (PM2) and port register 2 (P2) is not required when using the temperature sensor. There is no problem if the pin function is set as digital I/O.

**(1) Peripheral enable register 0 (PER0)**

Use the PER0 register in the same manner as during A/D converter basic operation (see 11.3 (1) Peripheral enable register 0 (PER0)).

**(2) A/D converter mode register (ADM)**

Use the ADM register in the same manner as during A/D converter basic operation (see 11.3 (2) A/D converter mode register (ADM)).

However, selection of the A/D conversion time when a temperature sensor is used varies as shown in Table 11-5.

**Table 11-5. Selection of A/D Conversion Time When Using Temperature Sensor**

**(1)  $2.7\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$**

A/D Converter Mode Register (ADM)					Conversion Time Selection			Conversion Clock ( $f_{AD}$ )	
FR2	FR1	FR0	LV1	LV0	$f_{CLK} = 2\text{ MHz}$	$f_{CLK} = 8\text{ MHz}$	$f_{CLK} = 20\text{ MHz}$		
0	0	0	0	1	$480/f_{CLK}$	Setting prohibited	$60.0\ \mu\text{s}$	$24.0\ \mu\text{s}$	$f_{CLK}/12$
0	0	1	0	1	$320/f_{CLK}$	Setting prohibited	$40.0\ \mu\text{s}$	Setting prohibited	$f_{CLK}/8$
0	1	0	0	1	$240/f_{CLK}$		$30.0\ \mu\text{s}$		$f_{CLK}/6$
0	1	1	0	1	$160/f_{CLK}$		Setting prohibited		$f_{CLK}/4$
1	0	0	0	1	$120/f_{CLK}$		$60.0\ \mu\text{s}$		$f_{CLK}/3$
1	0	1	0	1	$80/f_{CLK}$	$40.0\ \mu\text{s}$		$f_{CLK}/2$	
1	1	1	0	1	$40/f_{CLK}$	Setting prohibited		$f_{CLK}$	
Other than above					Setting prohibited				

**Cautions** 1. Set the conversion times so as to satisfy the following condition.

$f_{AD} = 0.6\text{ to }1.8\text{ MHz}$

2. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion (ADCS = 0) beforehand.
3. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

**Remark**  $f_{CLK}$ : CPU/peripheral hardware clock frequency

**(3) 10-bit A/D conversion result register (ADCR)**

Use the ADCR register in the same manner as during A/D converter basic operation (see 11.3 (3) 10-bit A/D conversion result register (ADCR)).

**Caution** When using a temperature sensor, use the result of the second or later A/D conversion for temperature sensor 0 (ANI0 side), and the result of the third or later A/D conversion for temperature sensor 1 (ANI1 side).

**(4) Analog input channel specification register (ADS)**

This register specifies the channel from which an analog voltage to be A/D-converted is input, in the same manner as during A/D converter basic operation. When a temperature sensor is used, however, some settings differ from those of A/D converter basic operation.

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

Reset signal generation clears this register to 00H.

**Figure 11-16. Format of Analog Input Channel Specification Register (ADS) When Using Temperature Sensor**

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	0	ADS3	ADS2	ADS1	ADS0

ADISS	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
1	0	0	0	0	ANI0	Temperature sensor 0 output
1	0	0	0	1	ANI1	Temperature sensor 1 output
Other than above					Setting prohibited	

**Caution** Be sure to clear bits 4 to 6 to “0”.

### 11.5.3 Temperature sensor operation

#### (1) Temperature sensor detection value

When using a temperature sensor, determine as reference temperatures two points of temperature (high and low) in the temperature range to be used, and measure the result of A/D conversion with temperature sensors 0 and 1 at each reference temperature in advance. Perform the measurement in the same environment as the one in which the temperature sensor is used in a set.

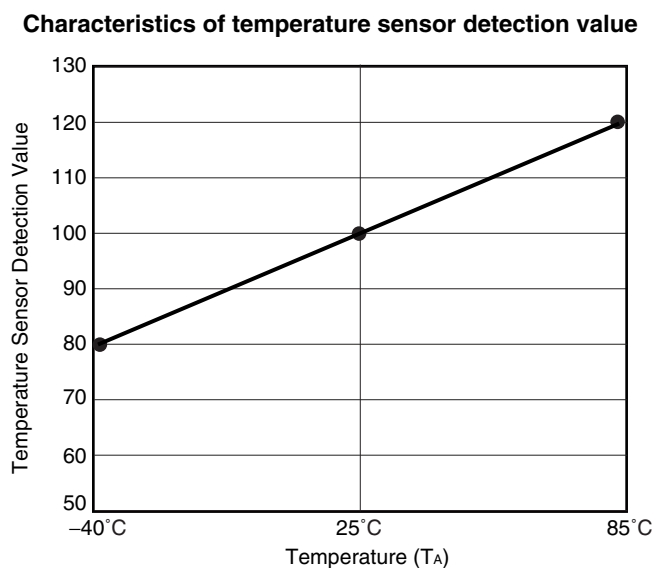
By using an expression of temperature sensor detection value characteristics, which are obtained from the values of high and low reference temperatures and the result of A/D conversion with temperature sensors 0 and 1 at an arbitrary temperature, the temperature at that time can be obtained.

**Remark** The value obtained from the ratio of the results of A/D conversion with a sensor that depends/does not depend on temperature is called a “temperature sensor detection value”.

- Sensor that depends on temperature  
Conversion channel: temperature sensor 0 (ANI0 side)  
A/D conversion result: ADT0
- Sensor that does not depend on temperature  
Conversion channel: temperature sensor 1 (ANI1 side)  
A/D conversion result: ADT1
- Temperature sensor detection value =  $KTV = \frac{ADT0}{ADT1} \times 256$

The characteristics (reference value) of the temperature sensor detection value are as follows.

**Figure 11-17. Characteristics of Temperature Sensor Detection Value (Reference Value)**





**(2) How to calculate temperature**

As shown in Figure 11-17, the temperature sensor detection value makes a characteristics curve that is linear with respect to the temperature. Therefore, the temperature sensor detection value can be expressed with the following expressions.

$$\text{Temperature sensor detection value} \cong \text{Tilt} \times (T_{\text{NOW}} - T_{\text{BASE1}}) + \text{Offset}$$

$$KTV_{\text{NOW}} \cong \frac{(KTV_{\text{BASE2}} - KTV_{\text{BASE1}})}{(T_{\text{BASE2}} - T_{\text{BASE1}})} \times (T_{\text{NOW}} - T_{\text{BASE1}}) + KTV_{\text{BASE1}}$$

$T_{\text{BASE1}}$ : Low reference temperature,  $T_{\text{BASE2}}$ : High reference temperature

$T_{\text{NOW}}$ : Temperature during sensor operation

$KTV_{\text{BASE1}}$ : Temperature sensor detection value at a low reference temperature

$KTV_{\text{BASE2}}$ : Temperature sensor detection value at a high reference temperature

$KTV_{\text{NOW}}$ : Temperature sensor detection value during temperature measurement

When  $ADT0_{\text{BASE1}}$ : Result of A/D conversion (sensor 0) at a low reference temperature

$ADT1_{\text{BASE1}}$ : Result of A/D conversion (sensor 1) at a low reference temperature

$ADT0_{\text{BASE2}}$ : Result of A/D conversion (sensor 0) at a high reference temperature

$ADT1_{\text{BASE2}}$ : Result of A/D conversion (sensor 1) at a high reference temperature

$ADT0_{\text{NOW}}$ : Result of A/D conversion (sensor 0) during temperature measurement

$ADT1_{\text{NOW}}$ : Result of A/D conversion (sensor 1) during temperature measurement

$KTV_{\text{BASE1}}$ ,  $KTV_{\text{BASE2}}$ , and  $KTV_{\text{NOW}}$  are obtained as follows.

$$KTV_{\text{BASE1}} = \frac{ADT0_{\text{BASE1}}}{ADT1_{\text{BASE1}}} \times 256$$

$$KTV_{\text{BASE2}} = \frac{ADT0_{\text{BASE2}}}{ADT1_{\text{BASE2}}} \times 256$$

$$KTV_{\text{NOW}} = \frac{ADT0_{\text{NOW}}}{ADT1_{\text{NOW}}} \times 256$$

Thus, temperature  $T_{\text{NOW}}$  is obtained by using the following expressions.

$$T_{\text{NOW}} \cong \frac{(KTV_{\text{NOW}} - KTV_{\text{BASE1}}) \times (T_{\text{BASE2}} - T_{\text{BASE1}})}{(KTV_{\text{BASE2}} - KTV_{\text{BASE1}})} + T_{\text{BASE1}}$$

$$T_{\text{NOW}} \cong \frac{ADT1_{\text{BASE2}} \times (ADT1_{\text{BASE1}} \times ADT0_{\text{NOW}} - ADT0_{\text{BASE1}} \times ADT1_{\text{NOW}}) \times (T_{\text{BASE2}} - T_{\text{BASE1}})}{ADT1_{\text{NOW}} \times (ADT1_{\text{BASE1}} \times ADT0_{\text{BASE2}} - ADT0_{\text{BASE1}} \times ADT1_{\text{BASE2}})} + T_{\text{BASE1}}$$

**Remarks 1.** When obtaining a temperature through calculation, it is recommended to determine the upper and lower end of the temperature range as the reference temperatures for measurement.

**2.** In addition to calculation, temperature  $T_{\text{NOW}}$  can also be obtained by measuring the temperature sensor detection values at each temperature in advance, preparing them as table data, and comparing them with the temperature sensor detection value during temperature measurement. With this method, table data must be created for each interval of temperatures to be detected.

### 11.5.4 Procedures for using temperature sensors

#### (1) Procedure for using temperature sensors

<1> Perform the following steps in the same environment as the one in which the temperature sensor is used in a set

- When obtaining a temperature through calculation
 

Determine as reference temperatures two points of temperature (high and low) in the temperature range to be used, and measure the result of A/D conversion with temperature sensors 0 and 1 at the reference temperature in advance, before shipment of the set.
- When obtaining a temperature through table reference
 

Measure the temperature sensor detection values at each temperature in advance, and prepare them as table data.

Store the above values into the internal flash memory area by means such as writing them via self programming, or store them into an external memory.

**Remark** When obtaining the temperature through calculation and the result of A/D conversion by temperature sensors 0 and 1 at a high and low temperature, it is recommended to determine the upper and lower end of the temperature range as the reference temperatures for measurement.

<2> To obtain a temperature, perform A/D conversion for the voltage output from temperature sensors 0 and 1 and calculation by using the expression based on ADT0 and ADT1, or calculate the temperature sensor detection value and compare it with table data prepared in advance.

#### (2) Procedure for obtaining ADT0 and ADT1 of temperature sensors 0 and 1

(ADT0<sub>BASE1</sub>, ADT1<sub>BASE1</sub>, ADT0<sub>BASE2</sub> and ADT1<sub>BASE2</sub> at reference temperatures, ADT0<sub>NOW</sub> and ADT1<sub>NOW</sub> during temperature measurement)

<Initial setting for A/D conversion>

- <1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1.
- <2> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.
- <3> Select the conversion time by using bits 5 to 1 (FR2 to FR0, LV1 and LV0) of ADM.

<Measurement by temperature sensor 0>

- <4> Set the analog input channel specification register (ADS) to "80H" to select temperature sensor 0.
- <5> Set bit 7 (ADCS) of ADM to 1 to start A/D conversion operation.
- <6> The first A/D conversion ends and an interrupt request signal (INTAD) occurs.
- <7> The second A/D conversion ends and an interrupt request signal (INTAD) occurs.
- <8> Read A/D conversion data (ADT0) from the A/D conversion result register (ADCR).

<Measurement by temperature sensor 1>

- <9> Set the analog input channel specification register (ADS) to "81H" to select temperature sensor 1.
- <10> The first A/D conversion ends and an interrupt request signal (INTAD) occurs.
- <11> The second A/D conversion ends and an interrupt request signal (INTAD) occurs.
- <12> The third A/D conversion ends and an interrupt request signal (INTAD) occurs.
- <13> Read A/D conversion data (ADT1) from the A/D conversion result register (ADCR).

(The procedure is continued on the next page.)

<Obtaining temperature  $T_{NOW}$ >

<14> Calculate the temperature by using either of the following methods.

- When obtaining a temperature through calculation  
During measurement at reference temperatures, write ADT0 and ADT1 to the internal flash memory by means such as self programming. During actual measurement, calculate the current temperature  $T_{NOW}$  by using the following expression, based on ADT0 and ADT1 at that time.

$$T_{NOW} \cong \frac{ADT1_{BASE2} \times (ADT1_{BASE1} \times ADT0_{NOW} - ADT0_{BASE1} \times ADT1_{NOW}) \times (T_{BASE2} - T_{BASE1})}{ADT1_{NOW} \times (ADT1_{BASE1} \times ADT0_{BASE2} - ADT0_{BASE1} \times ADT1_{BASE2})} + T_{BASE1}$$

- When obtaining a temperature through table reference  
Measure and calculate the temperature sensor detection values ( $ADT0/ADT1 \times 256$ ) based on ADT0 and ADT1 at each temperature interval. Set the temperature corresponding to that value as table data, and write it to the internal flash memory by means such as self programming.  
During actual measurement, calculate the temperature sensor detection value ( $ADT0/ADT1 \times 256$ ) based on ADT0 and ADT1 at that time, compare it with the value of table data, and obtain the current temperature  $T_{NOW}$ .

<Finishing A/D conversion>

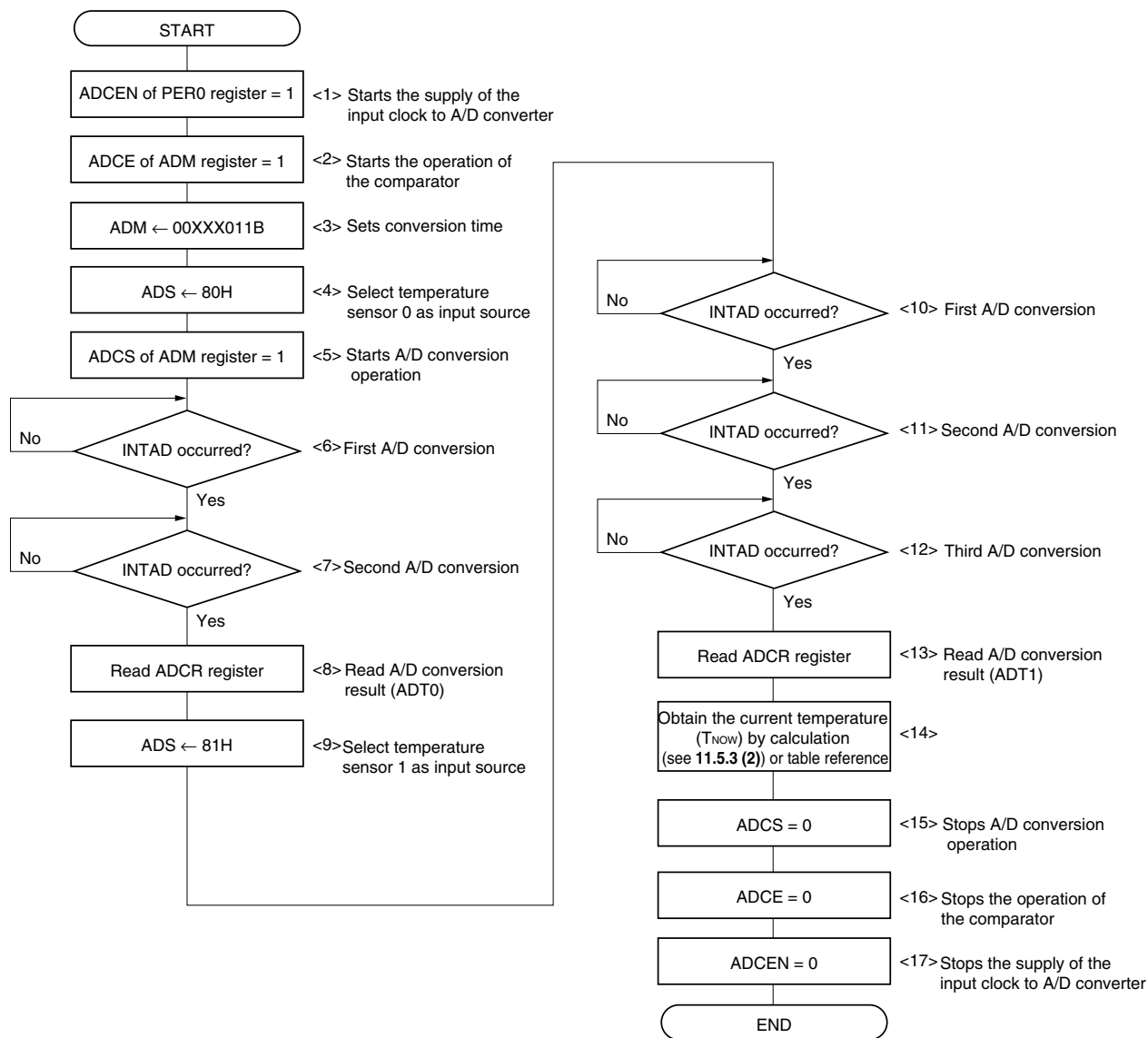
<15> Clear ADCS to 0.

<16> Clear ADCE to 0.

<17> Clear bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 0.

- Cautions**
1. Make sure the period of <2> to <5> is 1  $\mu$ s or more. If ADCS is set to 1 within 1  $\mu$ s, the result of the third and later conversion becomes valid on the sensor 0 side.
  2. <2> can be done between <3> and <4>.
  3. The period from <7> to <10> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM. The period from <9> to <10> is the conversion time set using FR2 to FR0, LV1, and LV0.
  4. Do not change the AVREF0 voltage during <4> to <13>. Although the temperature sensor detection value does not depend on the AVREF0 voltage and thus there is no problem even if the AVREF0 voltage varies at every temperature measurement, it must be stable during a measurement cycle (from <4> to <13>).

Figure 11-18. Flowchart of Procedure for Using Temperature Sensor



**Caution** Use the result of the second or later A/D conversion for temperature sensor 0 (ANI0 side), and the result of the third or later A/D conversion for temperature sensor 1 (ANI1 side).

**Remark** Steps <1> to <17> in Figure 11-18 correspond to steps <1> to <17> in 11.5.4 (2) Procedure for obtaining ADT0 and ADT1 of temperature sensors 0 and 1.

## 11.6 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

### (1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$$\begin{aligned} 1\text{LSB} &= 1/2^{10} = 1/1024 \\ &= 0.098\%\text{FSR} \end{aligned}$$

Accuracy has no relation to resolution, but is determined by overall error.

### (2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

### (3) Quantization error

When analog values are converted to digital values, a  $\pm 1/2\text{LSB}$  error naturally occurs. In an A/D converter, an analog input voltage in a range of  $\pm 1/2\text{LSB}$  is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 11-19. Overall Error

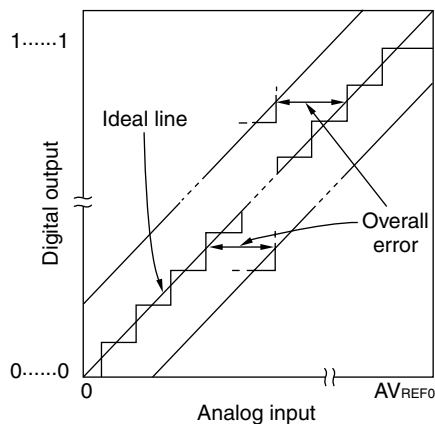
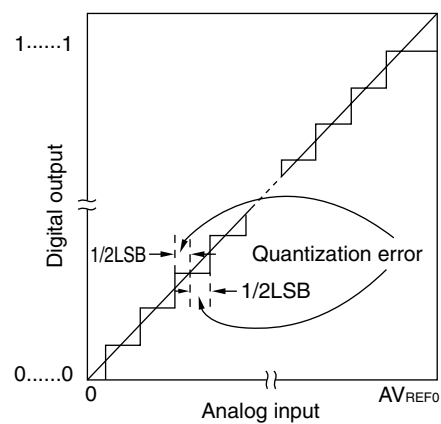


Figure 11-20. Quantization Error



### (4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value ( $1/2\text{LSB}$ ) 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 ( $3/2\text{LSB}$ ) when the digital output changes from  $0.....001$  to  $0.....010$ .

**(5) Full-scale error**

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

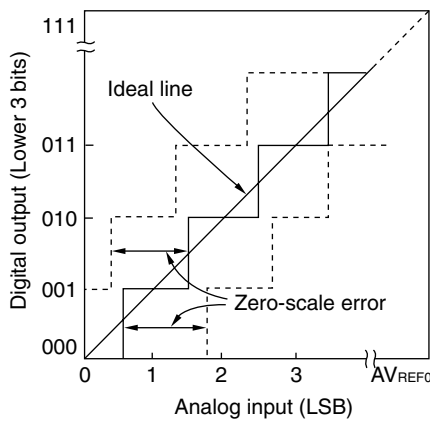
**(6) Integral linearity error**

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

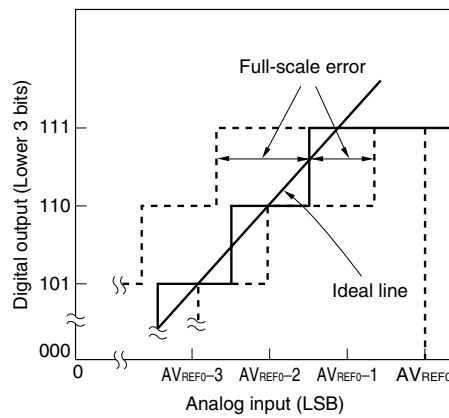
**(7) Differential linearity error**

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

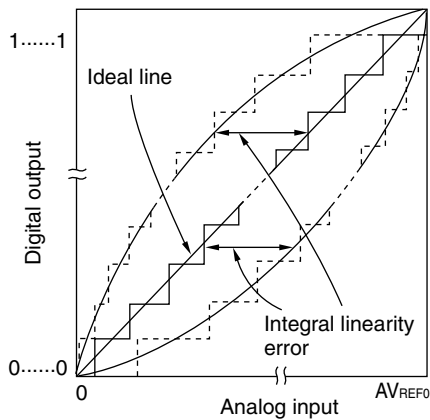
**Figure 11-21. Zero-Scale Error**



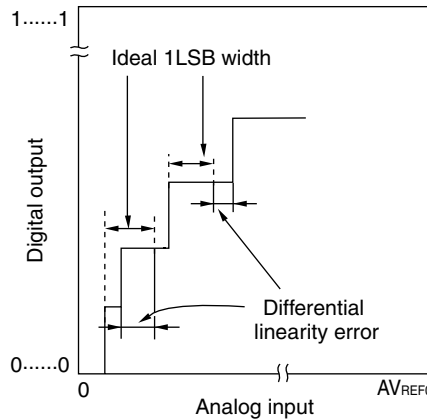
**Figure 11-22. Full-Scale Error**



**Figure 11-23. Integral Linearity Error**



**Figure 11-24. Differential Linearity Error**

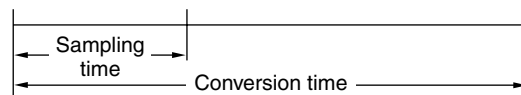


**(8) Conversion time**

This expresses the time from the start of sampling to when the digital output is obtained. The sampling time is included in the conversion time in the characteristics table.

**(9) Sampling time**

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



## 11.7 Cautions for A/D Converter

### (1) Operating current in STOP mode

Shift to STOP mode after clearing the A/D converter (by clearing bit 7 (ADCS) of the A/D converter mode register (ADM) to 0). The operating current can be reduced by clearing bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1L (IF1L) to 0 and start operation.

### (2) Reducing current when A/D converter is stopped

Be sure that the voltage to be applied to  $AV_{REF0}$  normally satisfies the conditions stated in Table 11-1.

If bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) are set to 0, the current will not be increased by the A/D converter even if a voltage is applied to  $AV_{REF0}$ , while the A/D converter is stopped. If a current flows from the power supply that supplies a voltage to  $AV_{REF0}$  to an external circuit of the microcontroller as shown in Figure 11-25,  $AV_{REF0} = 0\text{ V} = AV_{SS}$  can be achieved and the external current can be reduced by satisfying the following conditions.

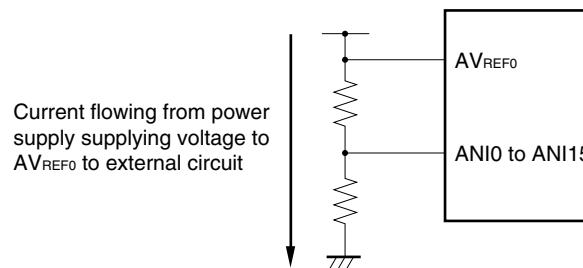
Set the following states before setting  $AV_{REF0} = 0\text{ V}$ .

- Set ADCS and ADCE of the A/D converter mode register (ADM) to 0.
- Set the port mode registers (PM20 to PM27 and PM150 to PM157) of the digital I/O pins to 1 to set to input mode, or set the digital I/O pins to low-level output (high-level output disabled) by setting the port mode registers (PM20 to PM27 and PM150 to PM157) and port registers (P20 to P27 and P150 to P157) to 0 to set to output mode.
- Make sure that no voltage is applied to all any of the analog or digital pins (P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15) (set to 0 V).

Do not perform the following operation when  $AV_{REF0} = 0\text{ V}$ .

- Do not access the port registers (P20 to P27 and P150 to P157) or port mode registers (PM20 to PM27 and PM150 to PM157) by using instructions or via DMA transfer.

**Figure 11-25. Example of Circuit Where Current Flows to External Circuit**



When restarting the A/D converter, operate it after the  $AV_{REF0}$  voltage rises and stabilizes and setting  $ADCE = 1$  (see **11.4.1 Basic operations of A/D converter** for the procedure for setting the A/D converter operation). Access digital ports after the  $AV_{REF0}$  voltage has risen and stabilized.

<R>

Stop the conversion performed by the D/A converter when the  $AV_{REF0}$  voltage is rising or falling.

**(3) Input range of ANI0 to ANI15**

Observe the rated range of the ANI0 to ANI15 input voltage. If a voltage of  $AV_{REF0}$  or higher and  $AV_{SS}$  or lower (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.

**(4) Conflicting operations**

<1> Conflict between A/D conversion result register (ADCR, ADCRH) write and ADCR or ADCRH read by instruction upon the end of conversion

ADCR or ADCRH read has priority. After the read operation, the new conversion result is written to ADCR or ADCRH.

<2> Conflict between ADCR or ADCRH write and A/D converter mode register (ADM) write, analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion

ADM, ADS, or ADPC write has priority. ADCR or ADCRH write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

**(5) Noise countermeasures**

To maintain the 10-bit resolution, attention must be paid to noise input to the  $AV_{REF0}$  pin and pins ANI0 to ANI15.

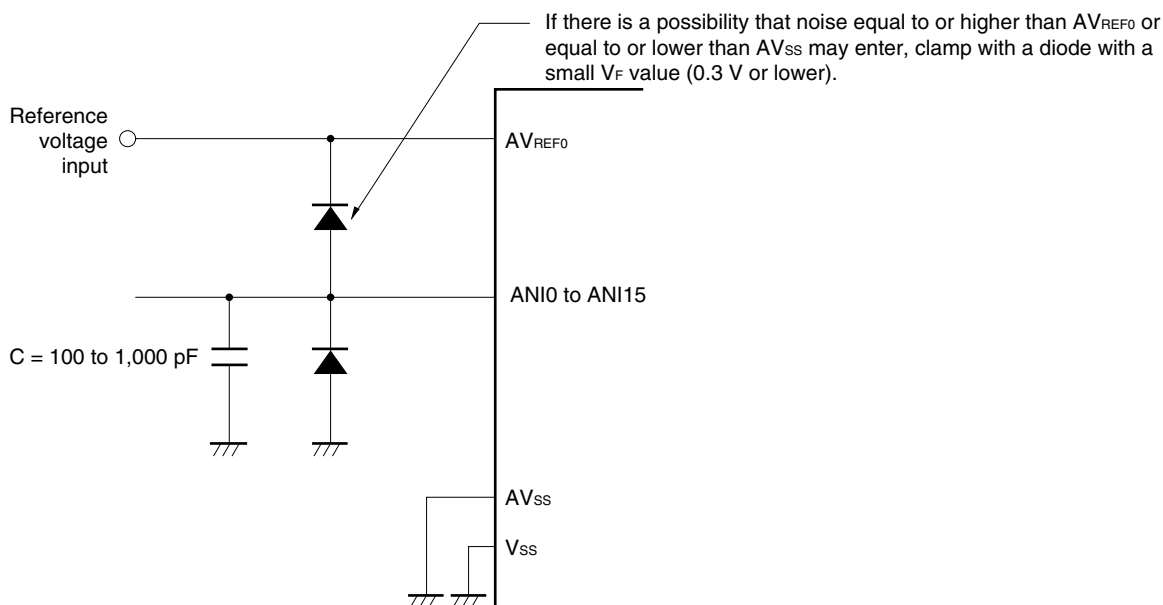
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<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 11-26 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

**Figure 11-26. Analog Input Pin Connection**





**(6) ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157**

- <1> The analog input pins (ANI0 to ANI7) are also used as input port pins (P20 to P27).  
The analog input pins (ANI8 to ANI15) are also used as input port pins (P150 to P157).  
When A/D conversion is performed with any of ANI0 to ANI15 selected, do not access P20 to P27 and P150 to P157 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to select pins used as P20 to P27 and P150 to P157 starting with the ANI0/P20 that is the furthest from AV<sub>REF0</sub>.
- <2> If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.

**(7) Input impedance of ANI0 to ANI15 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 10 k $\Omega$ , and to connect a capacitor of about 100 pF to the ANI0 to ANI15 pins (see **Figure 11-26**).

**(8) AV<sub>REF0</sub> pin input impedance**

A series resistor string of several tens of k $\Omega$  is connected between the AV<sub>REF0</sub> and AV<sub>SS</sub> pins.

Therefore, if the output impedance of the reference voltage source is high, this will result in a series connection to the series resistor string between the AV<sub>REF0</sub> and AV<sub>SS</sub> pins, resulting in a large reference voltage error.

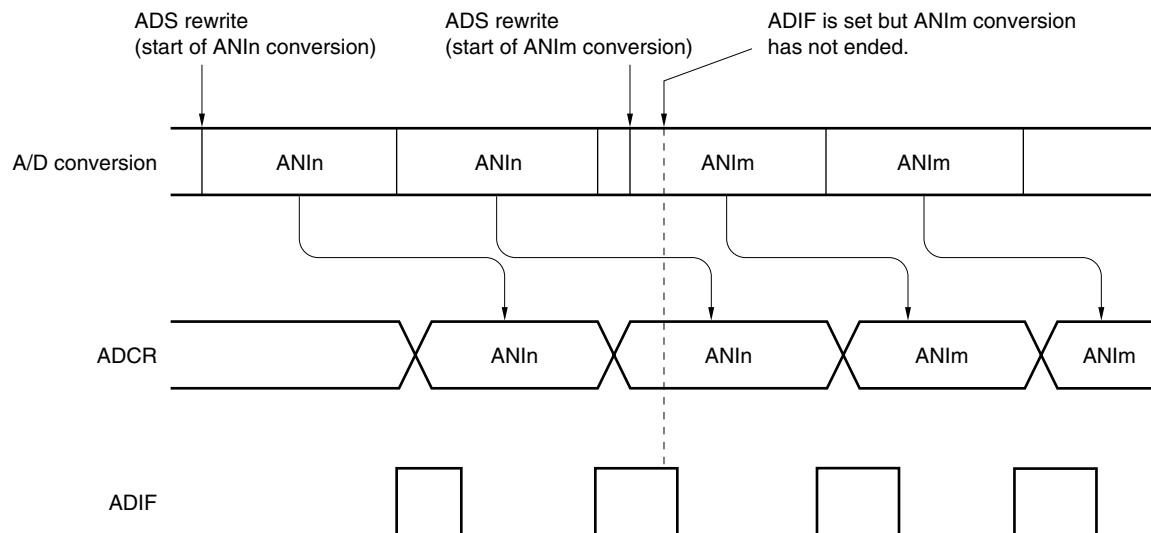
**(9) Interrupt request flag (ADIF)**

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF before the A/D conversion operation is resumed.

**Figure 11-27. Timing of A/D Conversion End Interrupt Request Generation**



- Remarks 1.** n = 0 to 15  
**2.** m = 0 to 15

**(10) 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 1  $\mu$ s 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.

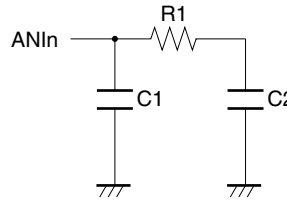
**(11) A/D conversion result register (ADCR, ADCRH) read operation**

When a write operation is performed to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR and ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using a timing other than the above may cause an incorrect conversion result to be read.

**(12) Internal equivalent circuit**

The equivalent circuit of the analog input block is shown below.

**Figure 11-28. Internal Equivalent Circuit of ANIn Pin**



**Table 11-6. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)**

$AV_{REF0}$	R1	C1	C2
$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	8.1 k $\Omega$	8 pF	5 pF
$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	31 k $\Omega$	8 pF	5 pF
$2.3\text{ V} \leq V_{DD} < 2.7\text{ V}$	381 k $\Omega$	8 pF	5 pF

**Remarks 1.** The resistance and capacitance values shown in Table 11-6 are not guaranteed values.

**2.** n = 0 to 15

<R>

**(13) Starting the A/D converter**

Start the A/D converter after the  $AV_{REF0}$  and  $AV_{REF1}$  voltages (the reference voltages for the D/A converter) stabilize.

## CHAPTER 12 D/A CONVERTER

### 12.1 Function of D/A Converter

The D/A converter has a resolution of 8 bits and converts an input digital signal into an analog signal. It is configured so that output analog signals of two channels (ANO0 and ANO1) can be controlled. The D/A converter has the following features.

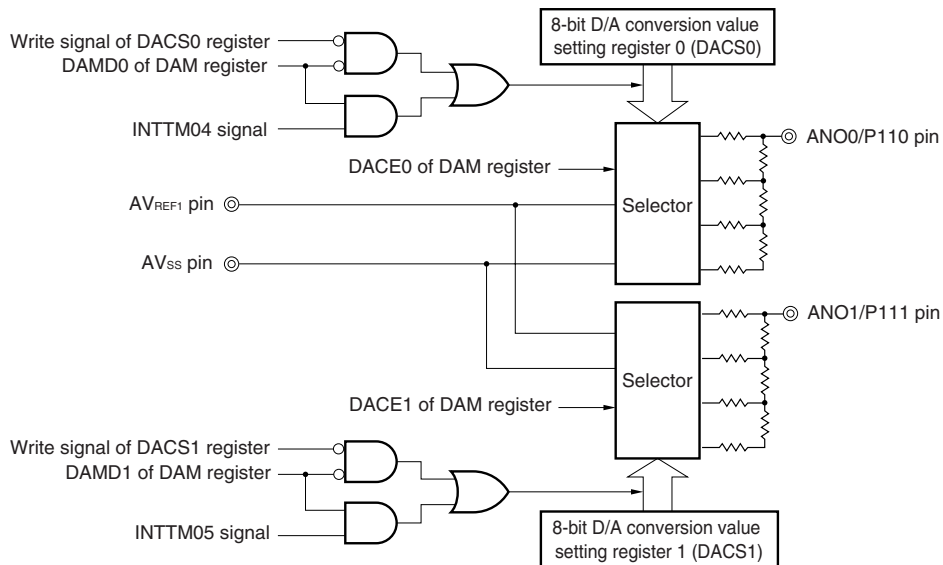
- 8-bit resolution × 2 channels
- R-2R ladder method
- Output analog voltage:  $AV_{REF1} \times m/256$  ( $AV_{REF1}$ : Reference voltage for D/A converter,  $m$ : Value set to DACSn register)
- Operation mode: Normal mode/real-time output mode

**Remark**  $n = 0, 1$

### 12.2 Configuration of D/A Converter

The configuration of the D/A converter is shown below.

**Figure 12-1. Block Diagram of D/A Converter**



- Remarks**
1. INTTM04 and INTTM05 are timer trigger signals (interrupt signals from timer channels 4 and 5) that are used in the real-time output mode.
  2. Channel 0 and channel 1 of the D/A converter share the  $AV_{REF1}$  pin.
  3. Channel 0 and channel 1 of the D/A converter share the  $AV_{SS}$  pin. The  $AV_{SS}$  pin is also shared with the A/D converter.

The D/A converter includes the following hardware.

**Table 12-1. Configuration of D/A Converter**

Item	Configuration
Control registers	Peripheral enable register 0 (PER0) D/A converter mode register (DAM) 8-bit D/A conversion value setting registers 0, 1 (DACS0, DACS1)

**(1) AVREF1 pin**

This is the D/A converter reference voltage input pin and the positive power supply pin of P110, P111, and the D/A converter.

The voltage that can be supplied to AVREF1 varies as follows, depending on whether the P110/ANO0 and P111/ANO1 pins are used as digital I/Os or analog outputs.

**Table 12-2. AVREF1 Voltage Applied to P110/ANO0 and P111/ANO1 Pins**

Analog/Digital	V <sub>DD</sub> Condition	AVREF1 Voltage
Using at least one pin as an analog output and using all pins not as digital I/Os	$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$1.8\text{ V} \leq AV_{REF1} \leq V_{DD} = EV_{DD0} = EV_{DD1}$
Pins used as analog outputs and digital I/Os are mixed <sup>Note</sup>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$2.7\text{ V} \leq AV_{REF1} \leq V_{DD} = EV_{DD0} = EV_{DD1}$
	$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	AVREF1 has same potential as EV <sub>DD0</sub> , EV <sub>DD1</sub> , and V <sub>DD</sub>
Using at least one pin as a digital I/O and using all pins not as analog outputs <sup>Note</sup>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$2.7\text{ V} \leq AV_{REF1} \leq V_{DD} = EV_{DD0} = EV_{DD1}$
	$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	AVREF1 has same potential as EV <sub>DD0</sub> , EV <sub>DD1</sub> , and V <sub>DD</sub>

**Note** AVREF1 is the reference for the I/O voltage of a port to be used as a digital port.

- High-/low-level input voltage (V<sub>IH5</sub>/V<sub>IL5</sub>)
- High-/low-level output voltage (V<sub>OH2</sub>/V<sub>OL2</sub>)

### 12.3 Registers Used in D/A Converter

The D/A converter uses the following registers.

- Peripheral enable register 0 (PER0)
- D/A converter mode register (DAM)
- 8-bit D/A conversion value setting registers 0, 1 (DACS0, DACS1)
- Port mode register 11 (PM11)
- Port register 11 (P11)

#### (1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the D/A converter is used, be sure to set bit 6 (DACEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

**Figure 12-2. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

DACEN	Control of D/A converter input clock
0	Stops supply of input clock. <ul style="list-style-type: none"> <li>• SFR used by the D/A converter cannot be written.</li> <li>• The D/A converter is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by the D/A converter can be read/written.</li> </ul>

**Caution** When setting the D/A converter, be sure to set DACEN to 1 first. If DACEN = 0, writing to a control register of the D/A converter is ignored, and, even if the register is read, only the default value is read (except for port mode register 11 (PM11) and port register 11 (P11)).

**(2) D/A converter mode register (DAM)**

This register controls the operation of the D/A converter.

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

Reset signal generation clears this register to 00H.

**Figure 12-3. Format of D/A Converter Mode Register (DAM)**

Address: FFF32H After reset: 00H R/W

Symbol	7	6	<5>	<4>	3	2	1	0
DAM	0	0	DACE1	DACE0	0	0	DAMD1	DAMD0

DACEn	Control of D/A conversion operation (n = 0, 1)
0	Stops conversion operation
1	Enables conversion operation

DAMDn	Selection of D/A converter operation mode (n = 0, 1)
0	Normal mode
1	Real-time output mode

**(3) 8-bit D/A conversion value setting registers 0 and 1 (DACS0, DACS1)**

These registers are used to set an analog voltage value to be output to the ANO0 and ANO1 pins.

DACS0 and DACS1 can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Figure 12-4. Format of 8-Bit D/A Conversion Value Setting Registers 0 and 1 (DACS0, DACS1)**

Address: FFF1CH, FFF1DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DACS <sub>n</sub>	DACS <sub>n</sub> 7	DACS <sub>n</sub> 6	DACS <sub>n</sub> 5	DACS <sub>n</sub> 4	DACS <sub>n</sub> 3	DACS <sub>n</sub> 2	DACS <sub>n</sub> 1	DACS <sub>n</sub> 0

**Remark** n = 0, 1

**(4) Port mode register 11 (PM11)**

This register sets the input or output of port 11 in 1-bit units.

When using the P110/ANO0 and P111/ANO1 pins as the analog output function of the D/A converter, set both PM110 and PM111 to 1. The output latches of P110 and P111 at this time may be 0 or 1.

PM11 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

**Figure 12-5. Format of Port Mode Register 11 (PM11)**

Address: FFF2BH    After reset: FFH    R/W

Symbol	7	6	5	4	3	2	1	0
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110

PM11n	P11n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)



## 12.4 Operation of D/A Converter

### 12.4.1 Operation in normal mode

D/A conversion is performed using write operation to the DACSn register as the trigger. The setting method is described below.

- <1> Set the DAMDn bit of the DAM register to 0 (normal mode).
- <2> Set the analog voltage value to be output to the ANOn pin to the DACSn register.  
Steps <1> and <2> above constitute the initial settings.
- <3> Set the DACEn bit of the DAM register to 1 (D/A conversion enable).  
D/A conversion starts and the analog voltage set in <2> is output to the ANOn pin when this setting is performed.  
The output level, however, is determined when the settling time elapses after D/A conversion starts.
- <4> To perform subsequent D/A conversions, write to the DACSn register.  
D/A conversion starts and an analog voltage is output to the ANOn pin when one  $f_{CLK}$  clock elapses after the write operation. The output level, however, is determined when the settling time elapses after D/A conversion starts.  
The previous D/A conversion result is held until the next D/A conversion is performed.  
When the DACEn bit of the DAM register is set to 0 (D/A conversion operation stop), analog voltage output is stopped, and the P110/ANO0 and P111/ANO1 pins can be used in port mode. At this time, the P110/ANO0 and P111/ANO1 pins are at high impedance because the PM11n bit of the PM11 register is 1 (input mode). The set value of the P11 register is output by setting the PM11n bit to 0 (output mode).

**Caution** Make the interval for writing DACSn of the same channel by one clock longer than  $f_{CLK}$ . If writing is successively performed, only the value written last will be converted.

- Remarks**
- 1.  $n = 0, 1$
  - 2.  $f_{CLK}$ : CPU/peripheral hardware clock

### 12.4.2 Operation in real-time output mode

D/A conversion is performed using the interrupt request signals (INTTM04 and INTTM05) of timer channel 4 and timer channel 5 as triggers.

The setting method is described below.

- <1> Set the DAMDn bit of the DAM register to 1 (real-time output mode).
- <2> Set the analog voltage value to be output to the ANOn pin to the DACSn register.
- <3> Set the DACE n bit of the DAM register to 1 (D/A conversion enable).

Steps <1> to <3> above constitute the initial settings.

- <4> Operate timer channel 4 and timer channel 5.
- <5> D/A conversion starts and the analog voltage set in <2> is output to the ANOn pin when the INTTM04 and INTTM05 signals are generated.

The output level, however, is determined when the settling time elapses after D/A conversion starts.

- <6> After that, the value set in the DACSn register is output every time the INTTM04 and INTTM05 signals are generated.

Set the analog voltage value to be output to the ANOn pin to the DACSn register before the next D/A conversion is started (INTTM04 and INTTM05 signals are generated).

When the DACE n bit of the DAM register is set to 0 (D/A conversion operation stop), analog voltage output is stopped, and the P110/ANO0 and P111/ANO1 pins can be used in port mode. At this time, the P110/ANO0 and P111/ANO1 pins are at high impedance because the PM11n bit of the PM11 register is 1 (input mode).

The set value of the P11 register is output by setting the PM11n bit to 0 (output mode).

D/A conversion starts by setting the DACE n bit, as described in <3>, and an analog voltage is output to the ANOn pin, but the output value of the ANOn pin up to <5> is undefined. An arbitrary value, however, can be output in <3> by performing the following settings before performing the setting in <1>.

- i. Set the DAMDn bit of the DAM register to 0 (normal mode).
- ii. Set the voltage value output from the ANOn pin in <3> to the DACSn register.
- iii. Afterward, perform <1> to <3>.

Consequently, the value set in ii can be output in <3>. The output level, however, is determined when the settling time elapses after D/A conversion starts.

**Cautions** 1. Make the interval for generating a start trigger to the same channel by one clock longer than  $f_{CLK}$ . If a start trigger is successively generated for every  $f_{CLK}$ , D/A conversion will be performed only at the first trigger.

- 2. Note the following points in the procedure (i to iii) for outputting an arbitrary value in <3>.
  - Do not generate the start trigger of the real-time output mode before enabling D/A conversion operation in <3> after the value is set to the DACSn register in ii.
  - An arbitrary value cannot be output in <3> if the DACE n bit of the PER0 register is cleared once after the value is set to the DACSn register in ii.

**Remarks** 1. For the output values of the ANO0 and ANO1 pins in the HALT and STOP modes, see **CHAPTER 19 STANDBY FUNCTION**.

- 2.  $n = 0, 1$
- 3.  $f_{CLK}$ : CPU/peripheral hardware clock

### 12.4.3 Cautions

Observe the following cautions when using the D/A converter of the 78K0R/KH3.

- <R>
- (1) The digital port I/O function, which is the alternate function of the ANO0 and ANO1 pins, does not operate during D/A conversion.  
During D/A conversion, 0 is read from the P11 register in input mode.
  - (2) Do not read/write the P11 register and do not change the setting of the PM11 register during D/A conversion (otherwise the conversion accuracy may decrease).
  - (3) It is recommended that both the ANO0 and ANO1 pins be used as analog output pins or digital I/O pins, that is, use these two channels for the same application (if these pins are used for the different applications, the conversion accuracy may decrease).
  - (4) In the real-time output mode, set the DACSn register value before the timer trigger is generated. In addition, do not change the set value of the DACSn register while the trigger signal is output.
  - (5) Before changing the operation mode, be sure to clear the DACEn bit of the DAM register to 0 (D/A conversion stop).
  - (6) When using the port that functions alternately as the ANO0 or ANO1 pin, use it as the port input with few level changes.
- <R>
- (7) Stop the conversion performed by the D/A converter when supplying  $AV_{REF1}$  or  $AV_{REF0}$  (the reference voltages for the A/D converter) starts or stops.
  - (8) Because the D/A converter stops operation in the STOP mode, the ANO0 and ANO1 pins go into a high-impedance state, and the power consumption can be reduced.  
In the standby modes other than the STOP mode, however, the operation continues. To lower the power consumption, therefore, clear the DACEn bit of the DAM register to 0 (D/A conversion stop).
  - (9) Since the output impedance of the D/A converter is high, the current cannot be obtained from the ANOn pin ( $n = 0, 1$ ). When the input impedance of the load is low, insert a follower amplifier between the load and ANOn pin keeping the wiring length as short as possible (for high impedance). If the wiring becomes too long, take necessary actions such as surrounding with a ground pattern.

## CHAPTER 13 SERIAL ARRAY UNIT

The serial array unit has four serial channels per unit and can use two or more of various serial interfaces (3-wire serial (CSI), UART, and simplified I<sup>2</sup>C) in combination.

Function assignment of each channel supported by the 78K0R/KH3 is as shown below (channels 2 and 3 of unit 1 are dedicated to UART3 (supporting LIN-bus)).

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I <sup>2</sup> C
0	0	CSI00	UART0	–
	1	CSI01		–
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

(Example of combination) When “UART0” is used for channels 0 and 1 of unit 0, CSI00 and CSI01 cannot be used, but CSI10, UART1, or IIC10 (CSI11, UART1, or IIC11) can be used.

### 13.1 Functions of Serial Array Unit

Each serial interface supported by the 78K0R/KH3 has the following features.

#### 13.1.1 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

This is a clocked communication function that uses three lines: serial clock ( $\overline{SCK}$ ) 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]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

### 13.1.2 UART (UART0, UART1, UART2, UART3)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. It transmits or receives data in asynchronization with the party of communication (by using an internal baud rate). Full-duplex UART communication can be realized by using two channels, one dedicated to transmission (even channel) and the other to reception (odd channel).

[Data transmission/reception]

- Data length of 5, 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

The LIN-bus is accepted in UART3 (2 and 3 channels of unit 1)

[LIN-bus functions]

- Wakeup signal detection
  - Sync break field (SBF) detection
  - Sync field measurement, baud rate calculation
- } External interrupt (INTPO) or timer array unit (TAU) is used.

### 13.1.3 Simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I<sup>2</sup>C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master and does not have a function to detect wait states.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function<sup>Note</sup> and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Parity error (ACK error)

\* [Functions not supported by simplified I<sup>2</sup>C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

**Note** An ACK is not output when the last data is being received by writing 0 to the SOEmn (SOEm register) bit and stopping the output of serial communication data. See **13.7.3 (2) Processing flow** for details.

**Remarks 1.** To use the full-function I<sup>2</sup>C bus, see **CHAPTER 14 SERIAL INTERFACE IIC0**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

## 13.2 Configuration of Serial Array Unit

Serial array unit includes the following hardware.

**Table 13-1. Configuration of Serial Array Unit**

Item	Configuration
Shift register	8 bits
Buffer register	Lower 8 bits of serial data register mn (SDRmn) <sup>Note</sup>
Serial clock I/O	SCK00, SCK01, SCK10, SCK11, SCK20, SCK21 pins (for 3-wire serial I/O), SCL10, SCL11, SCL20, SCL21 pins (for simplified I <sup>2</sup> C)
Serial data input	SI00, SI01, SI10, SI11, SI20, SI21 pins (for 3-wire serial I/O), RxD0, RxD1, RxD2 pins (for UART), RxD3 pin (for UART supporting LIN-bus)
Serial data output	SO00, SO01, SO10, SO11, SO20, SO21 pins (for 3-wire serial I/O), TxD0, TxD1, TxD2 pins (for UART), TxD3 pin (for UART supporting LIN-bus), output controller
Serial data I/O	SDA10, SDA11, SDA20, SDA21 pins (for simplified I <sup>2</sup> C)
Control registers	<p>&lt;Registers of unit setting block&gt;</p> <ul style="list-style-type: none"> <li>• Peripheral enable register 0 (PER0)</li> <li>• Serial clock select register m (SPSm)</li> <li>• Serial channel enable status register m (SEm)</li> <li>• Serial channel start register m (SSm)</li> <li>• Serial channel stop register m (STm)</li> <li>• Serial output enable register m (SOEm)</li> <li>• Serial output register m (SOM)</li> <li>• Serial output level register m (SOLm)</li> <li>• Input switch control register (ISC)</li> <li>• Noise filter enable register 0 (NFEN0)</li> </ul> <p>&lt;Registers of each channel&gt;</p> <ul style="list-style-type: none"> <li>• Serial data register mn (SDRmn)</li> <li>• Serial mode register mn (SMRmn)</li> <li>• Serial communication operation setting register mn (SCRmn)</li> <li>• Serial status register mn (SSRmn)</li> <li>• Serial flag clear trigger register mn (SIRmn)</li> <li>• Port input mode registers 0, 4, 9, 12, 14 (PIM0, PIM4, PIM9, PIM12, PIM14)</li> <li>• Port output mode registers 0, 4, 9, 12, 14 (POM0, POM4, POM9, POM12, POM14)</li> <li>• Port mode registers 0, 1, 4, 9, 12, 14 (PM0, PM1, PM4, PM9, PM12, PM14)</li> <li>• Port registers 0, 1, 4, 9, 12, 14 (P0, P1, P4, P9, P12, P14)</li> </ul>

**Note** The lower 8 bits of the serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

- CSIp communication ... SIOp (CSIp data register)
- UARTq reception ... RXDq (UARTq receive data register)
- UARTq transmission ... TXDq (UARTq transmit data register)
- IICr communication ... SIOr (IICr data register)

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3),

p: CSI number (p = 00, 01, 10, 11, 20, 21), q: UART number (q = 0 to 3), r: IIC number (r = 10, 11, 20, 21)

Figure 13-1 shows the block diagram of serial array unit 0.

Figure 13-1. Block Diagram of Serial Array Unit 0

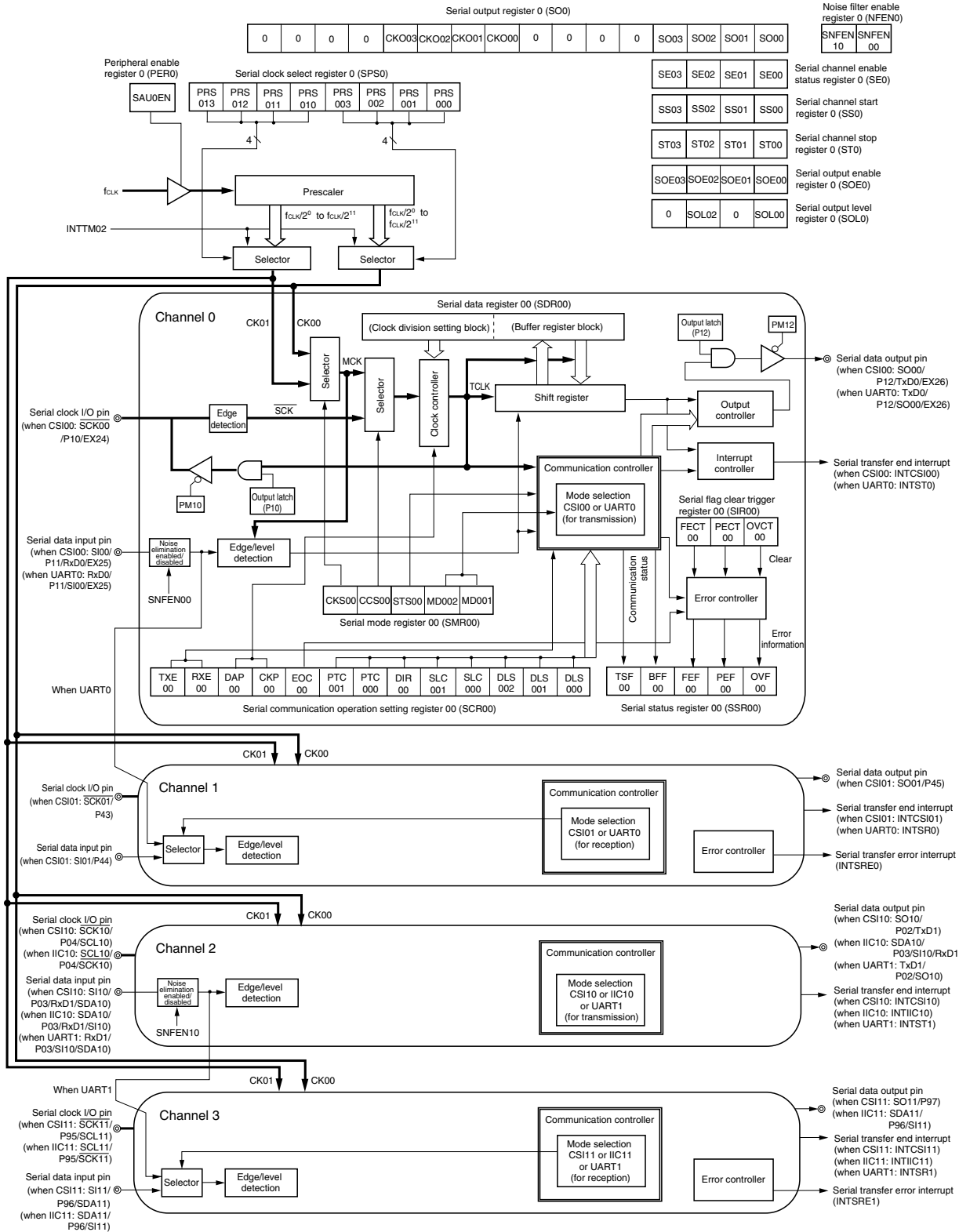
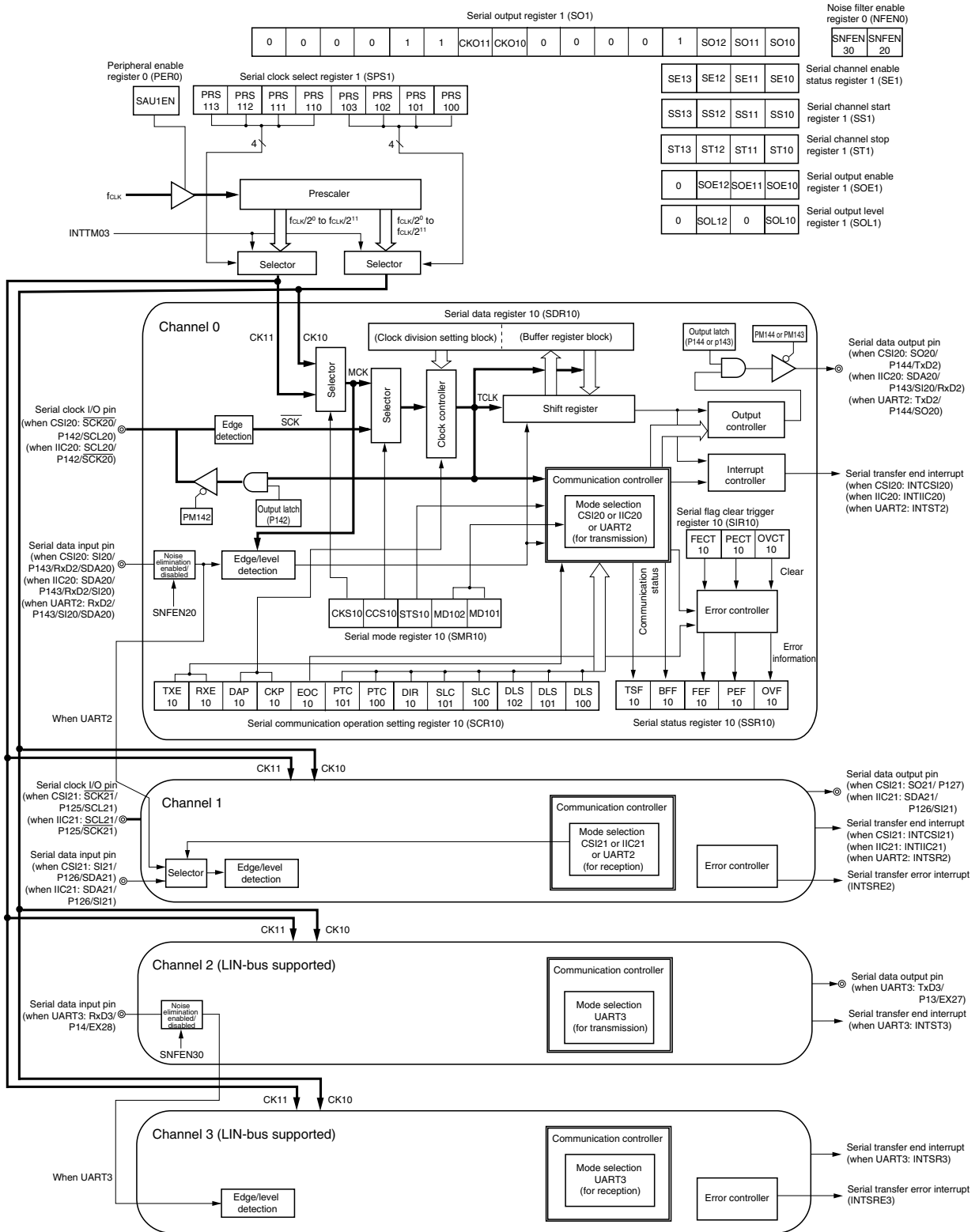




Figure 13-2 shows the block diagram of serial array unit 1.

Figure 13-2. Block Diagram of Serial Array Unit 1



**(1) Shift register**

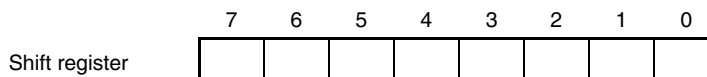
This is an 8-bit register that converts parallel data into serial data or vice versa.

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 8 bits of serial data register mn (SDRmn).

**(2) Lower 8 bits of the serial data register mn (SDRmn)**

SDRmn is the transmit/receive data register (16 bits) of channel n. Bits 7 to 0 function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (MCK).

When data is received, parallel data converted by the shift register is stored in the lower 8 bits. When data is to be transmitted, set transmit to be transferred to the shift register to the lower 8 bits.

The data stored in the lower 8 bits of this register is as follows, depending on the setting of bits 0 to 2 (DLSmn0 to DLSmn2) of the SCRmn register, regardless of the output sequence of the data.

- 5-bit data length (stored in bits 0 to 4 of SDRmn register) (settable in UART mode only)
- 7-bit data length (stored in bits 0 to 6 of SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of SDRmn register)

SDRmn can be read or written in 16-bit units.

The lower 8 bits of SDRmn of SDRmn can be read or written<sup>Note</sup> as the following SFR, depending on the communication mode.

- CSIp communication ... SIOp (CSIp data register)
- UARTq reception ... RXDq (UARTq receive data register)
- UARTq transmission ... TXDq (UARTq transmit data register)
- IICr communication ... SIOr (IICr data register)

**Note** Writing in 8-bit units is prohibited when the operation is stopped (SEmn = 0).

Reset signal generation clears this register to 0000H.

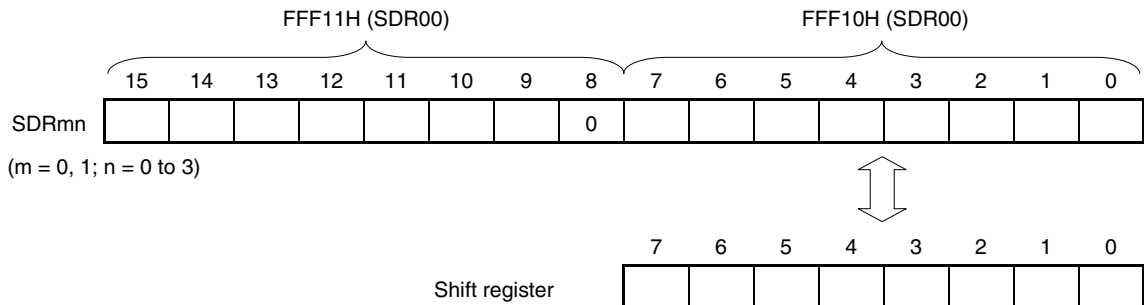
**Remarks 1.** After data is received, "0" is stored in bits 0 to 7 in bit portions that exceed the data length.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3),

p: CSI number (p = 00, 01, 10, 11, 20, 21), q: UART number (q = 0 to 3), r: IIC number (r = 10, 11, 20, 21)

**Figure 13-3. Format of Serial Data Register mn (SDRmn)**

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), After reset: 0000H R/W  
 FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),  
 FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11),  
 FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)



**Caution** Be sure to clear bit 8 to “0”.

- Remarks 1.** For the function of the higher 7 bits of SDRmn, see **13.3 Registers Controlling Serial Array Unit**.
- 2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3),

### 13.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial status register mn (SSRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial channel enable status register m (SEm)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOM)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 0, 4, 9, 12, 14 (PIM0, PIM4, PIM9, PIM12, PIM14)
- Port output mode registers 0, 4, 9, 12, 14 (POM0, POM4, POM9, POM12, POM14)
- Port mode registers 0, 1, 4, 9, 12, 14 (PM0, PM1, PM4, PM9, PM12, PM14)
- Port registers 0, 1, 4, 9, 12, 14 (P0, P1, P4, P9, P12, P14)

**Remark** m: Unit number (m = 0, 1)  
n: Channel number (n = 0 to 3)

**(1) Peripheral enable register 0 (PER0)**

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.

When serial array unit 1 is used, be sure to set bit 3 (SAU1EN) of this register to 1.

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

Reset signal generation clears this register to 00H.

**Figure 13-4. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

SAUmEN	Control of serial array unit m input clock
0	Stops supply of input clock. <ul style="list-style-type: none"> <li>• SFR used by serial array unit m cannot be written.</li> <li>• Serial array unit m is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by serial array unit m can be read/written.</li> </ul>

**Cautions 1.** When setting serial array unit m, be sure to set SAUmEN to 1 first. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read (except for input switch control register (ISC), noise filter enable register (NFEN0), port input mode registers (PIM0, PIM4, PIM9, PIM12, PIM14), port output mode registers (POM0, POM4, POM9, POM12, POM14), port mode registers (PM0, PM1, PM4, PM9, PM12, PM14), and port registers (P0, P1, P4, P9, P12, P14)).

**2.** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

**Remark** m: Unit number (m = 0, 1)

**(2) Serial clock select register m (SPSm)**

SPSm is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of SPSm, and CKm0 is selected by bits 3 to 0.

Rewriting SPSm is prohibited when the register is in operation (when SEMn = 1).

SPSm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SPSm can be set with an 8-bit memory manipulation instruction with SPSmL.

Reset signal generation clears this register to 0000H.

Figure 13-5. Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mp3	PRS mp2	PRS mp1	PRS mp0	f <sub>CLK</sub>	Section of operation clock (CKmp) <sup>Note 1</sup>				
					f <sub>CLK</sub> = 2 MHz	f <sub>CLK</sub> = 5 MHz	f <sub>CLK</sub> = 10 MHz	f <sub>CLK</sub> = 20 MHz	
0	0	0	0	f <sub>CLK</sub>	2 MHz	5 MHz	10 MHz	20 MHz	
0	0	0	1	f <sub>CLK</sub> /2	1 MHz	2.5 MHz	5 MHz	10 MHz	
0	0	1	0	f <sub>CLK</sub> /2 <sup>2</sup>	500 kHz	1.25 MHz	2.5 MHz	5 MHz	
0	0	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	250 kHz	625 kHz	1.25 MHz	2.5 MHz	
0	1	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	125 kHz	313 kHz	625 kHz	1.25 MHz	
0	1	0	1	f <sub>CLK</sub> /2 <sup>5</sup>	62.5 kHz	156 kHz	313 kHz	625 kHz	
0	1	1	0	f <sub>CLK</sub> /2 <sup>6</sup>	31.3 kHz	78.1 kHz	156 kHz	313 kHz	
0	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	
1	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	
1	0	0	1	f <sub>CLK</sub> /2 <sup>9</sup>	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	
1	0	1	0	f <sub>CLK</sub> /2 <sup>10</sup>	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	
1	0	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	
1	1	1	1	INTTM02 if m = 0, INTTM03 if m = 1 <sup>Note 2</sup>					
Other than above				Setting prohibited					

**Notes1.** When changing the clock selected for f<sub>CLK</sub> (by changing the system clock control register (CKC) value), do so after having stopped (STm = 000FH) the operation of the serial array unit (SAU). When selecting INTTM02 and INTTM03 for the operation clock, also stop the timer array unit 0 (TAU0) (TT0 = 00FFH).

**2.** SAU0 can be operated at a fixed division ratio of the subsystem clock, regardless of the f<sub>CLK</sub> frequency (main system clock, subsystem clock), by operating the interval timer for which f<sub>SUB</sub>/4 has been selected as the count clock (setting TIS02 (if m = 0) or TIS03 (if m = 1) of the TIS0 register to 1) and selecting INTTM02 and INTTM03 by using the SPSm register in channels 2 and 3 of TAU. When changing f<sub>CLK</sub>, however, SAU and TAU0 must be stopped as described in Note 1 above.

**Cautions 1.** Be sure to clear bits 15 to 8 to “0”.

**2.** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

**Remarks 1.** f<sub>CLK</sub>: CPU/peripheral hardware clock frequency

f<sub>SUB</sub>: Subsystem clock frequency

**2.** m: Unit number (m = 0, 1), p = 0, 1

**(3) Serial mode register mn (SMRmn)**

SMRmn is a register that sets an operation mode of channel n. It is also used to select an operation clock (MCK), specify whether the serial clock ( $\overline{SCK}$ ) may be input or not, set a start trigger, an operation mode (CSI, UART, or I<sup>2</sup>C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting SMRmn is prohibited when the register is in operation (when SEMn = 1). However, the MDmn0 bit can be rewritten during operation.

SMRmn can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets this register to 0020H.

**Figure 13-6. Format of Serial Mode Register mn (SMRmn) (1/2)**

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), After reset: 0020H R/W  
 F0150H, F0151H (SMR10), F0152H, F0153H (SMR11),  
 F0154H, F0155H (SMR12), F0156H, F0157H (SMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn	0	SIS mn0	1	0	0	MD mn2	MD mn1	MD mn0

CKS mn	Selection of operation clock (MCK) of channel n
0	Operation clock CKm0 set by SPSm register
1	Operation clock CKm1 set by SPSm register
Operation clock MCK is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (TCLK) is generated.	

CCS mn	Selection of transfer clock (TCLK) of channel n
0	Divided operation clock MCK specified by CKSmn bit
1	Clock input from $\overline{SCK}$ pin (slave transfer in CSI mode)
Transfer clock TCLK is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of MCK is set by the higher 7 bits of the SDRmn register.	

STS mn	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I <sup>2</sup> C).
1	Valid edge of RxD pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

**Caution** Be sure to clear bits 13 to 9, 7, 4, and 3 to “0”. Be sure to set bit 5 to “1”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 13-6. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), After reset: 0020H R/W  
 F0150H, F0151H (SMR10), F0152H, F0153H (SMR11),  
 F0154H, F0155H (SMR12), F0156H, F0157H (SMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn	0	SIS mn0	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0	Controls inversion of level of receive data of channel n in UART mode	
0	Falling edge is detected as the start bit. The input communication data is captured as is.	
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.	

MD mn2	MD mn1	Setting of operation mode of channel n
0	0	CSI mode
0	1	UART mode
1	0	Simplified I <sup>2</sup> C mode
1	1	Setting prohibited

MD mn0	Selection of interrupt source of channel n	
0	Transfer end interrupt	
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)	
For successive transmission, the next transmit data is written by setting MDmn0 to 1 when SDRmn data has run out.		

<R>

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)



**(4) Serial communication operation setting register mn (SCRmn)**

SCRmn is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting SCRmn is prohibited when the register is in operation (when SEMn = 1).

SCRmn can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets this register to 0087H.

**Figure 13-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/3)**

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), After reset: 0087H R/W  
 F0158H, F0159H (SCR10), F015AH, F015BH (SCR11),  
 F015CH, F015DH (SCR12), F015EH, F015FH (SCR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Does not start communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	Selection of data and clock phase in CSI mode	Type
0	0	$\overline{SCKp}$  $SOp$ Slp input timing	1
0	1	$\overline{SCKp}$  $SOp$ Slp input timing	2
1	0	$\overline{SCKp}$  $SOp$ Slp input timing	3
1	1	$\overline{SCKp}$  $SOp$ Slp input timing	4

Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I<sup>2</sup>C mode.

**Caution** Be sure to clear bits 3, 6, and 11 to “0”. Be sure to set bit 2 to “1”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

Figure 13-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), After reset: 0087H R/W  
 F0158H, F0159H (SCR10), F015AH, F015BH (SCR11),  
 F015CH, F015DH (SCR12), F015EH, F015FH (SCR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

EOC mn	Selection of masking of error interrupt signal (INTSREx (x = 0 to 3))
0	Masks error interrupt INTSREx (INTSRx is not masked).
1	Enables generation of error interrupt INTSREx (INTSRx is masked if an error occurs).
Set EOCmn = 0 in the CSI mode, simplified I <sup>2</sup> C mode, and during UART transmission <sup>Note 1</sup> . Set EOCmn = 1 during UART reception.	

PTC mn1	PTC mn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity <sup>Note 2</sup> .	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the CSI mode and simplified I <sup>2</sup> C mode.			

DIR mn	Selection of data transfer sequence in CSI and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIRmn = 0 in the simplified I <sup>2</sup> C mode.	

**Notes 1.** When using CSI01 not with EOC01 = 0, error interrupt INTSRE0 may be generated.  
 When using CSI11 or IIC11 not with EOC03 = 0, error interrupt INTSRE1 may be generated.  
 When using CSI21 or IIC21 not with EOC11 = 0, error interrupt INTSRE2 may be generated.  
**2.** 0 is always added regardless of the data contents.

<R>

**Caution** Be sure to clear bits 3, 6, and 11 to “0”. Be sure to set bit 2 to “1”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

**Figure 13-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (3/3)**

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), After reset: 0087H R/W  
 F0158H, F0159H (SCR10), F015AH, F015BH (SCR11),  
 F015CH, F015DH (SCR12), F015EH, F015FH (SCR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

SLC mn1	SLC mn0	Setting of stop bit in UART mode
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits
1	1	Setting prohibited

When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred.  
 Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I<sup>2</sup>C mode.  
 Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the CSI mode.

DLS mn2	DLS mn1	DLS mn0	Setting of data length in CSI and UART modes
1	0	0	5-bit data length (stored in bits 0 to 4 of SDRmn register) (settable in UART mode only)
1	1	0	7-bit data length (stored in bits 0 to 6 of SDRmn register)
1	1	1	8-bit data length (stored in bits 0 to 7 of SDRmn register)
Other than above			Setting prohibited

Be sure to set DLSmn0 = 1 in the simplified I<sup>2</sup>C mode.

**Caution** Be sure to clear bits 3, 6, and 11 to “0”. Be sure to set bit 2 to “1”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

**(5) Higher 7 bits of the serial data register mn (SDRmn)**

SDRmn is the transmit/receive data register (16 bits) of channel n. Bits 7 to 0 function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (MCK). If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by the higher 7 bits of SDRmn is used as the transfer clock.

For the function of the lower 8 bits of SDRmn, see **13.2 Configuration of Serial Array Unit**.

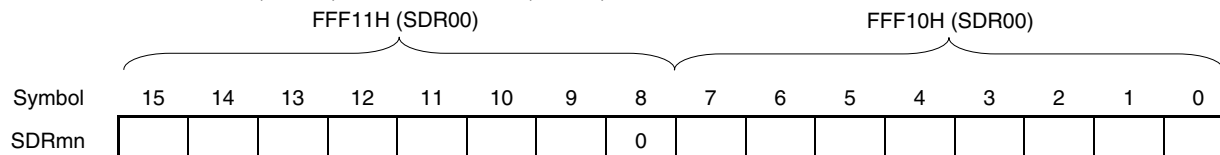
SDRmn can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8 bits of SDRmn. When SDRmn is read during operation, 0 is always read.

Reset signal generation clears this register to 0000H.

**Figure 13-8. Format of Serial Data Register mn (SDRmn)**

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), After reset: 0000H R/W  
 FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),  
 FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11),  
 FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)



SDRmn[15:9]							Setting of division ratio of operation clock (MCK)
0	0	0	0	0	0	0	MCK/2
0	0	0	0	0	0	1	MCK/4
0	0	0	0	0	1	0	MCK/6
0	0	0	0	0	1	1	MCK/8
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	MCK/254
1	1	1	1	1	1	1	MCK/256

**Cautions 1.** Be sure to clear bit 8 to “0”.

**2.** Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

<R> **3.** Setting SDRmn[15:9] = 0000000B is prohibited when simplified I<sup>2</sup>C is used. Set SDRmn[15:9] to 0000001B or greater.

<R> **4.** Do not write eight bits to the lower eight bits if operation is stopped (SEmn = 0). (If these bits are written to, the higher seven bits are cleared to 0.)

**Remarks 1.** For the function of the lower 8 bits of SDRmn, see **13.2 Configuration of Serial Array Unit**.

**2.** m: Unit number (m = 0, 1)

n: Channel number (n = 0 to 3)

**(6) Serial status register mn (SSRmn)**

SSRmn is a register that indicates the communication status and error occurrence status of channel n. The errors indicated by this register are a framing error, parity error, and overrun error.

SSRmn can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of SSRmn can be set with an 8-bit memory manipulation instruction with SSRmnL.

Reset signal generation clears this register to 0000H.

**Figure 13-9. Format of Serial Status Register mn (SSRmn) (1/2)**

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), After reset: 0000H R  
 F0140H, F0141H (SSR10), F0142H, F0143H (SSR11),  
 F0144H, F0145H (SSR12), F0146H, F0147H (SSR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn	PEF mn	OVF mn

TSF mn	Communication status indication flag of channel n
0	Communication is not under execution.
1	Communication is under execution.
Because this flag is an updating flag, it is automatically cleared when the communication operation is completed. This flag is cleared also when the STmn/SSmn bit is set to 1.	

BFF mn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
This is an updating flag. It is automatically cleared when transfer from the SDRmn register to the shift register is completed. During reception, it is automatically cleared when data has been read from the SDRmn register. This flag is cleared also when the STmn/SSmn bit is set to 1.	
This flag is automatically set if transmit data is written to the SDRmn register when the TXEmn bit of the SCRmn register = 1 (transmission or reception mode in each communication mode). It is automatically set if receive data is stored in the SDRmn register when the RXEmn bit of the SCRmn register = 1 (transmission or reception mode in each communication mode). It is also set in case of a reception error.	
If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVFmn = 1) is detected.	

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 13-9. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), After reset: 0000H R  
 F0140H, F0141H (SSR10), F0142H, F0143H (SSR11),  
 F0144H, F0145H (SSR12), F0146H, F0147H (SSR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn	PEF mn	OVF mn

FEF mn	Framing error detection flag of channel n
0	No error occurs.
1	A framing error occurs during UART reception. <Framing error cause> A framing error occurs if the stop bit is not detected upon completion of UART reception.
This is a cumulative flag and is not cleared until 1 is written to the FECTmn bit of the SIRmn register.	

PEF mn	Parity error detection flag of channel n
0	Error does not occur.
1	A parity error occurs during UART reception or ACK is not detected during I <sup>2</sup> C transmission. <Parity error cause> <ul style="list-style-type: none"> <li>• A parity error occurs if the parity of transmit data does not match the parity bit on completion of UART reception.</li> <li>• ACK is not detected if the ACK signal is not returned from the slave in the timing of ACK reception during I<sup>2</sup>C transmission.</li> </ul>
This is a cumulative flag and is not cleared until 1 is written to the PECTmn bit of the SIRmn register.	

OVF mn	Overrun error detection flag of channel n
0	No error occurs.
1	An overrun error occurs. <Causes of overrun error> <ul style="list-style-type: none"> <li>• Receive data stored in the SDRmn register is not read and transmit data is written or the next receive data is written.</li> <li>• Transmit data is not ready for slave transmission or reception in the CSI mode.</li> </ul>
This is a cumulative flag and is not cleared until 1 is written to the OVCTmn bit of the SIRmn register.	

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

**(7) Serial flag clear trigger register mn (SIRmn)**

SIRmn is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFMn) of serial status register mn is cleared to 0. Because SIRmn is a trigger register, it is cleared immediately when the corresponding bit of SSRmn is cleared.

SIRmn can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SIRmn can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears this register to 0000H.

**Figure 13-10. Format of Serial Flag Clear Trigger Register mn (SIRmn)**

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03), After reset: 0000H R/W  
 F0148H, F0149H (SIR10), F014AH, F014BH (SIR11),  
 F014CH, F014DH (SIR12), F014EH, F014FH (SIR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FEC Tmn	PEC Tmn	OVC Tmn

FEC Tmn	Clear trigger of framing error of channel n
0	No trigger operation
1	Clears the FEFmn bit of the SSRmn register to 0.

PEC Tmn	Clear trigger of parity error flag of channel n
0	No trigger operation
1	Clears the PEFmn bit of the SSRmn register to 0.

OVC Tmn	Clear trigger of overrun error flag of channel n
0	No trigger operation
1	Clears the OVFMn bit of the SSRmn register to 0.

**Caution** Be sure to clear bits 15 to 3 to “0”.

- Remarks**
1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
  2. When the SIRmn register is read, 0000H is always read.

**(8) Serial channel enable status register m (SEm)**

SEm indicates whether data transmission/reception operation of each channel is enabled or stopped. When 1 is written a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1. When 1 is written a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0. Channel n that is enabled to operate cannot rewrite by software the value of CKOmn of the serial output register m (SOM) to be described below, and a value reflected by a communication operation is output from the serial clock pin. Channel n that stops operation can set the value of CKOmn of the SOM register by software and output its value from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software. SEm can be read by a 16-bit memory manipulation instruction. The lower 8 bits of SEm can be set with an 1-bit or 8-bit memory manipulation instruction with SEmL. Reset signal generation clears this register to 0000H.

**Figure 13-11. Format of Serial Channel Enable Status Register m (SEm)**

Address: F0120H, F0121H (SE0), F0160H, F0161H (SE1) After reset: 0000H R

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

SEm n	Indication of operation enable/stop status of channel n
0	Operation stops (stops with the values of the control register and shift register, and the statuses of the serial clock I/O pin, serial data output pin, and the FEF, PEF, and OVF error flags retained <sup>Note</sup> ).
1	Operation is enabled.

**Note** Bits 6 and 5 (TSFmn, BFFmn) of the SSRmn register are cleared.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)



**(9) Serial channel start register m (SSm)**

SSm is a trigger register that is used to enable starting communication/count by each channel.  
 When 1 is written a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1. Because SSmn is a trigger bit, it is cleared immediately when SEmn = 1.  
 SSm can be set by a 16-bit memory manipulation instruction.  
 The lower 8 bits of SSm can be set with an 1-bit or 8-bit memory manipulation instruction with SSmL.  
 Reset signal generation clears this register to 0000H.

**Figure 13-12. Format of Serial Channel Start Register m (SSm)**

Address: F0122H, F0123H (SS0), F0162H, F0163H (SS1) After reset: 0000H R/W

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

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets SEmn to 1 and enters the communication wait status (if a communication operation is already under execution, the operation is stopped and the start condition is awaited).

**Caution** Be sure to clear bits 15 to 4 to “0”.

- Remarks**
1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
  2. When the SSm register is read, 0000H is always read.

**(10) Serial channel stop register m (STm)**

STm is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (STmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0. Because STmn is a trigger bit, it is cleared immediately when SEmn = 0.

STm can set written by a 16-bit memory manipulation instruction.

The lower 8 bits of STm can be set with an 1-bit or 8-bit memory manipulation instruction with STmL.

Reset signal generation clears this register to 0000H.

**Figure 13-13. Format of Serial Channel Stop Register m (STm)**

Address: F0124H, F0125H (ST0), F0164H, F0165H (ST1) After reset: 0000H R/W

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

STm n	Operation stop trigger of channel n
0	No trigger operation
1	Clears SEmn to 0 and stops the communication operation. (Stops with the values of the control register and shift register, and the statuses of the serial clock I/O pin, serial data output pin, and the FEF, PEF, and OVF error flags retained <sup>Note</sup> .)

**Note** Bits 6 and 5 (TSFmn, BFFmn) of the SSRmn register are cleared.

**Caution** Be sure to clear bits 15 to 4 to “0”.

**Remarks** 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)  
2. When the STm register is read, 0000H is always read.

**(11) Serial output enable register m (SOEm)**

SOEm is a register that is used to enable or stop output of the serial communication operation of each channel.

Channel n that enables serial output cannot rewrite by software the value of SOMn of the serial output register m (SOM) to be described below, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOMn value of the SOM register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform of the start condition and stop condition can be created by software.

SOEm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SOEm can be set with an 1-bit or 8-bit memory manipulation instruction with SOEmL.

Reset signal generation clears this register to 0000H.

**Figure 13-14. Format of Serial Output Enable Register m (SOEm)**

Address: F012AH, F012BH    After reset: 0000H    R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	SOE 03	SOE 02	SOE 01	SOE 00

Address: F016AH, F016BH    After reset: 0000H    R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE 12	SOE 11	SOE 10

SOE mn	Serial output enable/disable of channel n															
0	Stops output by serial communication operation.															
1	Enables output by serial communication operation.															

**Caution** Be sure to clear bits 15 to 4 of SOE0, and bits 15 to 3 of SOE1 to “0”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3),  
mn = 00 to 03, 10 to 12

**(12) Serial output register m (SOM)**

SOM is a buffer register for serial output of each channel.

The value of bit n of this register is output from the serial data output pin of channel n.

The value of bit (n + 8) of this register is output from the serial clock output pin of channel n.

SOMn of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

CKOMn of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of CKOMn can be changed only by a serial communication operation.

To use the P02/SO10/TxD1, P03/SI10/SDA10/RxD1, P04/SCK10/SCL10, P10/SCK00/EX24, P12/SO00/TxD0/EX26, P13/TxD3/EX27, P43/SCK01, P45/SO01, P95/SCK11/SCL11, P96/SI11/SDA11, P97/SO11, P125/SCK21/SCL21, P126/SI21/SDA21, P127/SO21, P142/SCK20/SCL20, P143/SI20/SDA20/RxD2, or P144/SO20/TxD2 pin as a port function pin, set the corresponding CKOMn and SOMn bits to "1".

SOM can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0F0FH.

**Figure 13-15. Format of Serial Output Register m (SOM)**

Address: F0128H, F0129H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S00	0	0	0	0	CKO 03	CKO 02	CKO 01	CKO 00	0	0	0	0	SO 03	SO 02	SO 01	SO 00

Address: F0168H, F0169H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S01	0	0	0	0	1	1	CKO 11	CKO 10	0	0	0	0	1	SO 12	SO 11	SO 10

CKO mn	Serial clock output of channel n
0	Serial clock output value is "0".
1	Serial clock output value is "1".

SO mn	Serial data output of channel n
0	Serial data output value is "0".
1	Serial data output value is "1".

**Caution** Be sure to set bits 11, 10, and 3 of S01 to "1". And be sure to clear bits 15 to 12 and 7 to 4 of SOM to "0".

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 12

**(13) Serial output level register m (SOLm)**

SOLm is a register that is used to set inversion of the data output level of each channel.

This register can be set only in the UART mode. Be sure to set 0000H in the CSI mode and simplifies I<sup>2</sup>C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting SOLm is prohibited when the register is in operation (when SEMn = 1).

SOLm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of SOLm can be set with an 8-bit memory manipulation instruction with SOLmL.

Reset signal generation clears this register to 0000H.

**Figure 13-16. Format of Serial Output Level Register m (SOLm)**

Address: F0134H, F0135H (SOL0), F0174H, F0175H (SOL1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL m2	0	SOL m0

SOL mn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

**Caution** Be sure to clear bits 15 to 3 and 1 to “0”.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

**(14) Input switch control register (ISC)**

ISC is used to realize a LIN-bus communication operation by UART3 in coordination with an external interrupt and the timer array unit.

When bit 0 is set to 1, the input signal of the serial data input (RxD3) pin is selected as an external interrupt (INTP0) that can be used to detect a wakeup signal.

When bit 1 is set to 1, the input signal of the serial data input (RxD3) pin is selected as a timer input, so that the pulse widths of a sync break field and a sync field can be measured by the timer.

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

Reset signal generation clears this register to 00H.

**Figure 13-17. Format of Input Switch Control Register (ISC)**

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

<R>

ISC1	Switching channel 7 input of timer array unit
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of RxD3 pin is used as timer input (to measure the pulse widths of the sync break field and sync field).

<R>

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD3 pin as an external interrupt (wakeup signal detection).

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

**(15) Noise filter enable register 0 (NFEN0)**

NFEN0 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 or simplified I<sup>2</sup>C communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is enabled, CPU/peripheral operating clock (f<sub>CLK</sub>) is synchronized with 2-clock match detection.

NFEN0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

**Figure 13-18. Format of Noise Filter Enable Register 0 (NFEN0)**

Address: F0060H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	SNFEN30	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN30	Use of noise filter of RxD3/P14 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN30 to 1 to use the RxD3 pin. Clear SNFEN30 to 0 to use the P14 pin.	

SNFEN20	Use of noise filter of RxD2/SDA20/SI20/P143 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN20 to 1 to use the RxD2 pin. Clear SNFEN20 to 0 to use the SDA20, SI20, and P143 pins.	

SNFEN10	Use of noise filter of RxD1/SDA10/SI10/P03 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN10 to 1 to use the RxD1 pin. Clear SNFEN10 to 0 to use the SDA10, SI10, and P03 pins.	

SNFEN00	Use of noise filter of RxD0/SI00/P11 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN00 to 1 to use the RxD0 pin. Clear SNFEN00 to 0 to use the SI00 and P11 pins.	

**Caution** Be sure to clear bits 7, 5, 3, and 1 to “0”.

**(16) Port input mode registers 0, 4, 9, 12, 14 (PIM0, PIM4, PIM9, PIM12, PIM14)**

These registers set the input buffer of ports 0, 4, 9, 12, and 14 in 1-bit units.

PIM0, PIM4, PIM9, PIM12, and PIM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Figure 13-19. Format of Port Input Mode Registers 0, 4, 9, 12, and 14 (PIM0, PIM4, PIM9, PIM12, PIM14)**

Address F0040H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
PIM0	0	0	0	PIM04	PIM03	0	0	0

Address F0044H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
PIM4	0	0	0	PIM44	PIM43	0	0	0

Address F0049H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
PIM9	0	PIM96	PIM95	0	0	0	0	0

Address F004CH    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
PIM12	0	PIM126	PIM125	0	0	0	0	0

Address F004EH    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
PIM14	0	0	0	0	PIM143	PIM142	0	0

PIMmn	Pmn pin input buffer selection (m = 0, 4, 9, 12, 14; n = 2 to 6)
0	Normal input buffer
1	TTL input buffer



**(17) Port output mode registers 0, 4, 9, 12, 14 (POM0, POM4, POM9, POM12, POM14)**

These registers set the output mode of ports 0, 4, 9, 12, and 14 in 1-bit units.

POM0, POM4, POM9, POM12, and POM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Figure 13-20. Format of Port Output Mode Registers 0, 4, 9, 12, and 14 (POM0, POM4, POM9, POM12, POM14)**

Address F0050H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
POM0	0	0	0	POM04	POM03	POM02	0	0

Address F0054H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
POM4	0	0	POM45	0	POM43	0	0	0

Address F0059H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
POM9	POM97	POM96	POM95	0	0	0	0	0

Address F005CH    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
POM12	POM127	POM126	POM125	0	0	0	0	0

Address F005EH    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
POM14	0	0	0	POM144	POM143	POM142	0	0

POMmn	Pmn pin output buffer selection (m = 0, 4, 9, 12, 14; n = 2 to 7)
0	Normal output mode
1	N-ch open-drain output (V <sub>DD</sub> tolerance) mode

**(18) Port mode registers 0, 1, 4, 9, 12, 14 (PM0, PM1, PM4, PM9, PM12, PM14)**

These registers set input/output of ports 0, 1, 4, 9, 12, and 14 in 1-bit units.

When using the P02/SO10/TxD1, P03/SI10/RxD1/SDA10, P04/SCK10/SCL10, P10/SCK00/EX24, P12/SO00/TxD0/EX26, P13/TxD3/EX27, P43/SCK01, P45/SO01, P95/SCK11/SCL11, P96/SI11/SDA11, P97/SO11, P125/SCK21/SCL21, P126/SI21/SDA21, P127/SO21, P142/SCK20/SCL20, P143/SI20/RxD2/SDA20, and P144/SO20/TxD2 pins for serial data output or serial clock output, clear the PM02 to PM04, PM10, PM12, PM13, PM43, PM45, PM95 to PM97, PM125 to PM127, and PM142 to PM144 bits to 0, and set the output latches of P02 to P04, P10, P12, P13, P43, P45, P95 to P97, P125 to P127, and P142 to P144 to 1.

When using the P03/SI10/RxD1/SDA10, P04/SCK10/SCL10, P10/SCK00/EX24, P11/SI00/RxD0/EX25, P14/RxD3/EX28, P43/SCK01, P44/SI01, P95/SCK11/SCL11, P96/SI11/SDA11, P125/SCK21/SCL21, P126/SI21/SDA21, P142/SCK20/SCL20, and P143/SI20/RxD2/SDA20 pins for serial data input or serial clock input, set the PM03, PM04, PM10, PM11, PM14, PM43, PM44, PM95, PM96, PM125, PM126, PM142, and PM143 bits to 1. At this time, the output latches of P03, P04, P10, P11, P14, P43, P44, P95, P96, P125, P126, P142, and P143 may be 0 or 1.

PM0, PM1, PM4, PM9, PM12, and PM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

**Figure 13-21. Format of Port Mode Registers 0, 1, 4, 9, 12, and 14 (PM0, PM1, PM4, PM9, PM12, PM14)**

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

Address: FFF29H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90

Address: FFF2CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM12	PM127	PM126	PM125	1	1	1	1	PM120

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140

PMmn	Pmn pin I/O mode selection (m = 0, 1, 4, 9, 12, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

### 13.4 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 P02/SO10/TxD1, P03/SI10/SDA10/RxD1, P04/SCK10/SCL10, P10/SCK00/EX24, P11/SI00/RxD0/EX25, P12/SO00/TxD0/EX26, P13/TxD3/EX27, P14/RxD3/EX28, P43/SCK01, P44/SI01, P45/SO01, P95/SCK11/SCL11, P96/SI11/SDA11, P97/SO11, P125/SCK21/SCL21, P126/SI21/SDA21, P127/SO21, P142/SCK20/SCL20, P143/SI20/SDA20/RxD2, or P144/SO20/TxD2 pin can be used as ordinary port pins in this mode.

#### 13.4.1 Stopping the operation by units

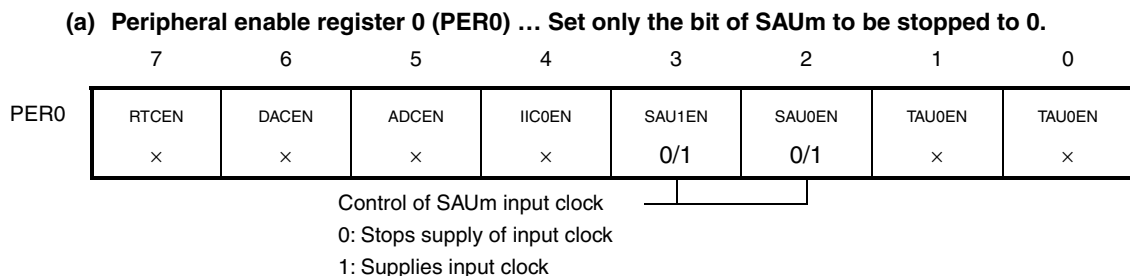
The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0.

To stop the operation of serial array unit 1, set bit 3 (SAU1EN) to 0.

**Figure 13-22. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units**



**Caution** If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read (except for input switch control register (ISC), noise filter enable register (NFEN0), port input mode registers (PIM0, PIM4, PIM9, PIM12, PIM14), port output mode registers (POM0, POM4, POM9, POM12, POM14), port mode registers (PM0, PM1, PM4, PM9, PM12, PM14), and port registers (P0, P1, P4, P9, P12, P14)).

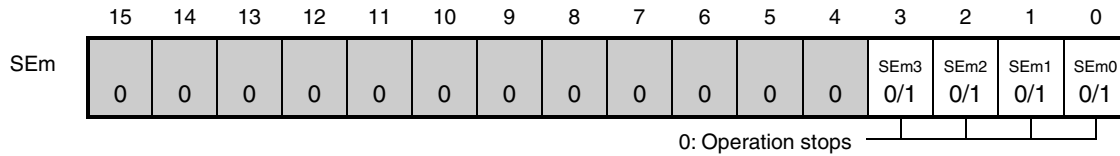
**Remark** m: Unit number (m = 0, 1)  
 x: Bits not used with serial array units (depending on the settings of other peripheral functions)  
 0/1: Set to 0 or 1 depending on the usage of the user

13.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

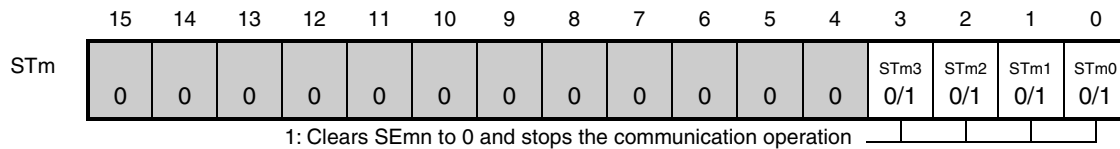
Figure 13-23. Each Register Setting When Stopping the Operation by Channels (1/2)

(a) **Serial Channel Enable Status Register m (SEm) ... This register indicates whether data transmission/reception operation of each channel is enabled or stopped.**



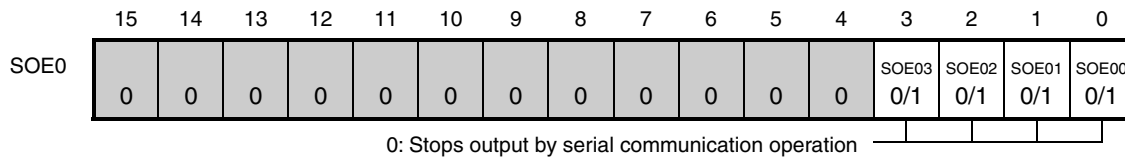
\* The SEm register is a read-only status register, whose operation is stopped by using the STm register.  
With a channel whose operation is stopped, the value of CKOm<sub>n</sub> of the SOm register can be set by software.

(b) **Serial channel stop register m (STm) ... This register is a trigger register that is used to enable stopping communication/count by each channel.**

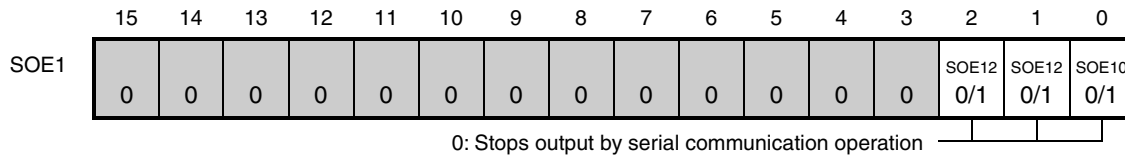


\* Because STm<sub>n</sub> is a trigger bit, it is cleared immediately when SEm<sub>n</sub> = 0.

(c) **Serial output enable register m (SOEm) ... This register is a register that is used to enable or stop output of the serial communication operation of each channel.**



\* For channel n, whose serial output is stopped, the SO0<sub>n</sub> value of the SO0 register can be set by software.



\* For channel n, whose serial output is stopped, the SO1<sub>n</sub> value of the SO1 register can be set by software.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

■ : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

Figure 13-23. Each Register Setting When Stopping the Operation by Channels (2/2)

(d) Serial output register m (SOm) ...This register is a buffer register for serial output of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0					CKO03	CKO02	CKO01	CKO00					SO03	SO02	SO01	SO00
	0	0	0	0	0/1	0/1	0/1	0/1	0	0	0	0	0/1	0/1	0/1	0/1

1: Serial clock output value is "1" 1: Serial data output value is "1"

\* When using pins corresponding to each channel as port function pins, set the corresponding CKO0n and SO0n bits to "1".

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1							CKO11	CKO10					SO12	SO11	SO10	
	0	0	0	0	1	1	0/1	0/1	0	0	0	0	1	0/1	0/1	0/1

1: Serial clock output value is "1" 1: Serial data output value is "1"

\* When using pins corresponding to each channel as port function pins, set the corresponding CKO1n and SO1n bits to "1".

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

: Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

### 13.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) Communication

This is a clocked communication function that uses three lines: serial clock ( $\overline{SCK}$ ) 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]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

The channels supporting 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) are channels 0 to 3 of SAU0 and channel 0 and 1 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I <sup>2</sup> C
0	0	CSI00	UART0	–
	1	CSI01		–
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

3-wire serial I/O (CSI00, CSI01, CIS10, CIS11, CSI20, CIS21) performs the following six types of communication operations.

- Master transmission (See 13.5.1.)
- Master reception (See 13.5.2.)
- Master transmission/reception (See 13.5.3.)
- Slave transmission (See 13.5.4.)
- Slave reception (See 13.5.5.)
- Slave transmission/reception (See 13.5.6.)

### 13.5.1 Master transmission

Master transmission is that the 78K0R/KH3 outputs a transfer clock and transmits data to another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK11, SO11	SCK20, SO20	SCK21, SO21
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.					
Error detection flag	None					
Transfer data length	7 or 8 bits					
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] <sup>Note</sup> $f_{CLK}$ : System clock frequency					
Data phase	Selectable by DAPmn bit <ul style="list-style-type: none"> <li>• DAPmn = 0: Data output starts from the start of the operation of the serial clock.</li> <li>• DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.</li> </ul>					
Clock phase	Selectable by CKPmn bit <ul style="list-style-type: none"> <li>• CKPmn = 0: Forward</li> <li>• CKPmn = 1: Reverse</li> </ul>					
Data direction	MSB or LSB first					

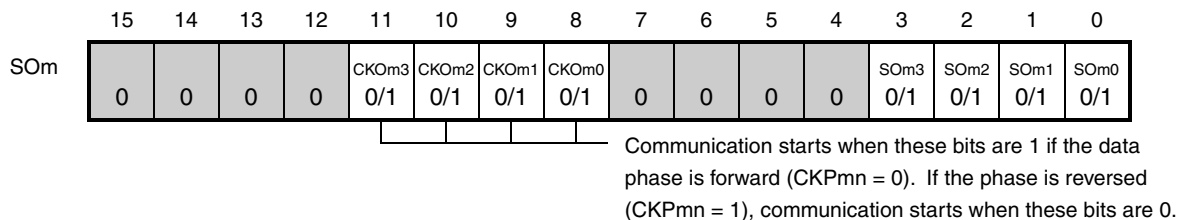
**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)** and **CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)**).

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

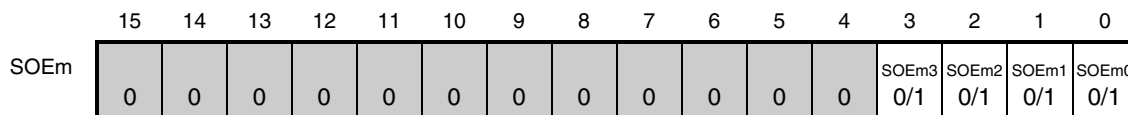
(1) Register setting

Figure 13-24. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

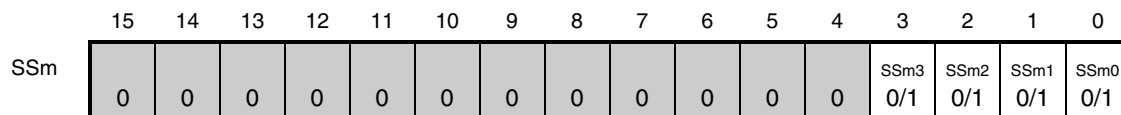
(a) Serial output register m (SOM) ... Sets only the bits of the target channel.



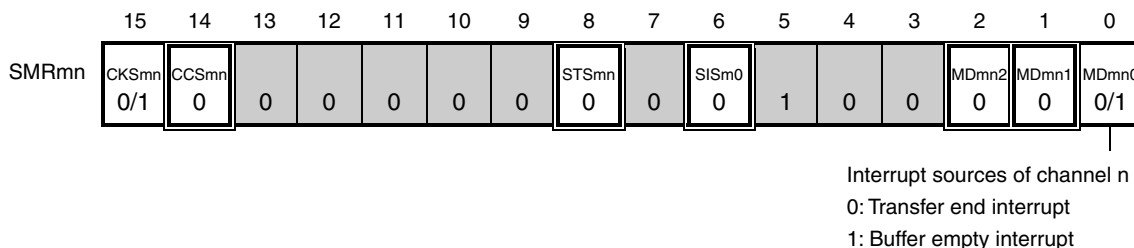
(b) Serial output enable register m (SOEm) ... Sets only the bits of the target channel to 1.



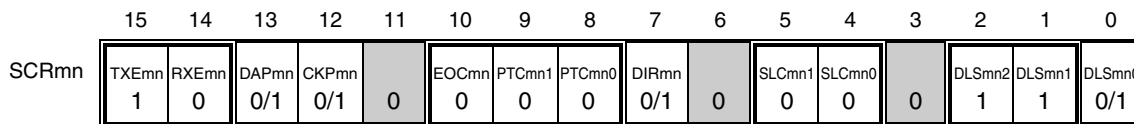
(c) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.



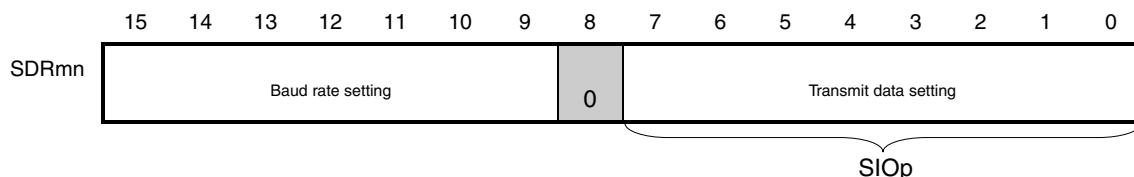
(d) Serial mode register mn (SMRmn)



(e) Serial communication operation setting register mn (SCRmn)



(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

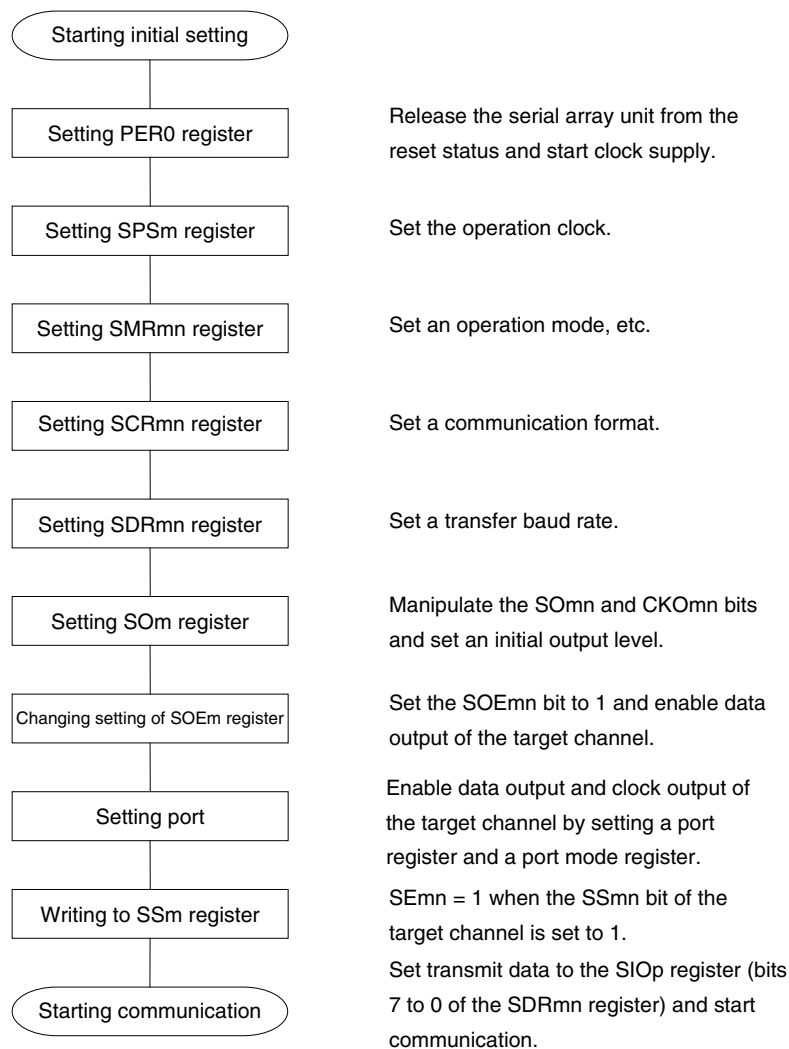


**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)  
: Setting is fixed in the CSI master transmission mode, : Setting disabled (set to the initial value)  
 0/1: Set to 0 or 1 depending on the usage of the user

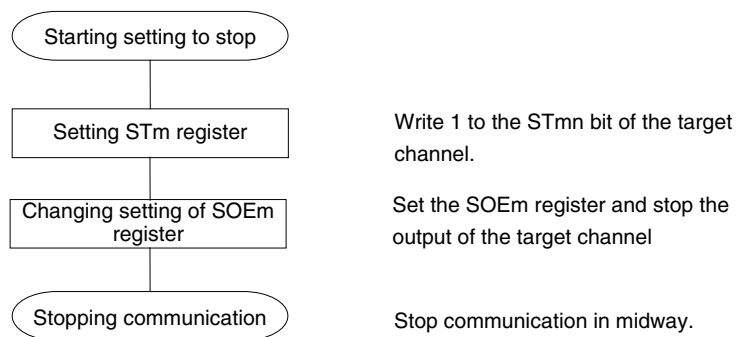


## (2) Operation procedure

Figure 13-25. Initial Setting Procedure for Master Transmission

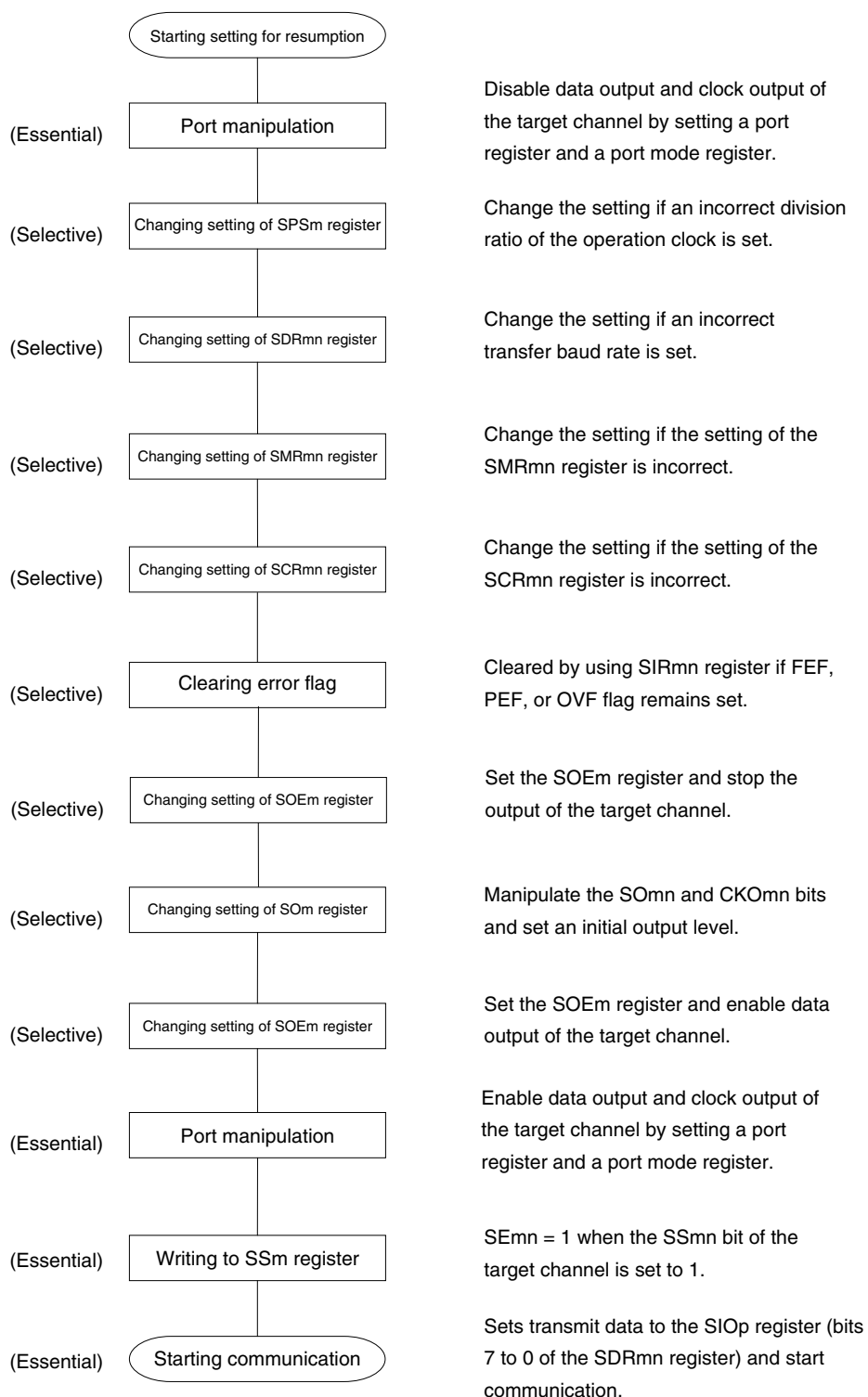


**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

**Figure 13-26. Procedure for Stopping Master Transmission**

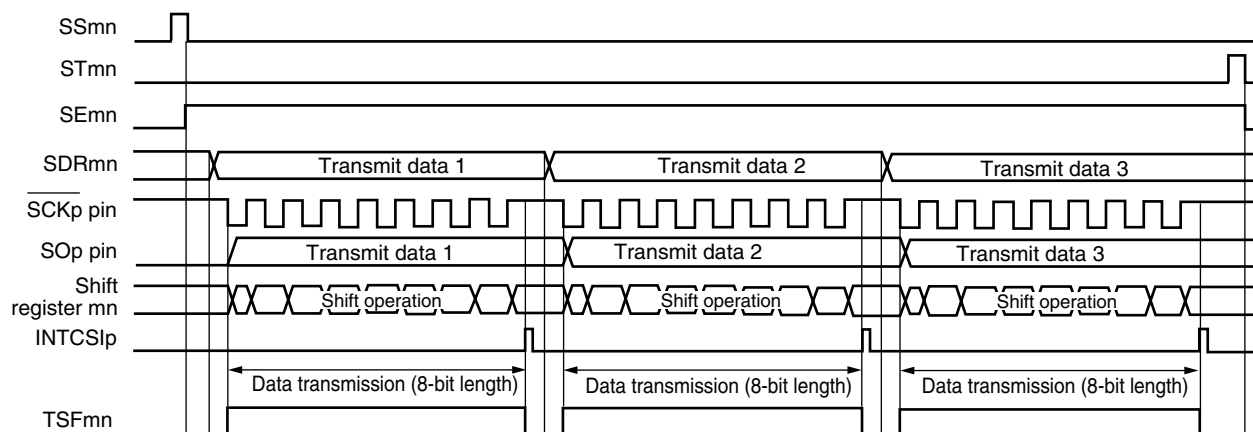
- Remarks 1.** Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 13-27 Procedure for Resuming Master Transmission**).
- 2.** p: CSI number (p = 00, 01, 10, 11, 20, 21)

**Figure 13-27. Procedure for Resuming Master Transmission**



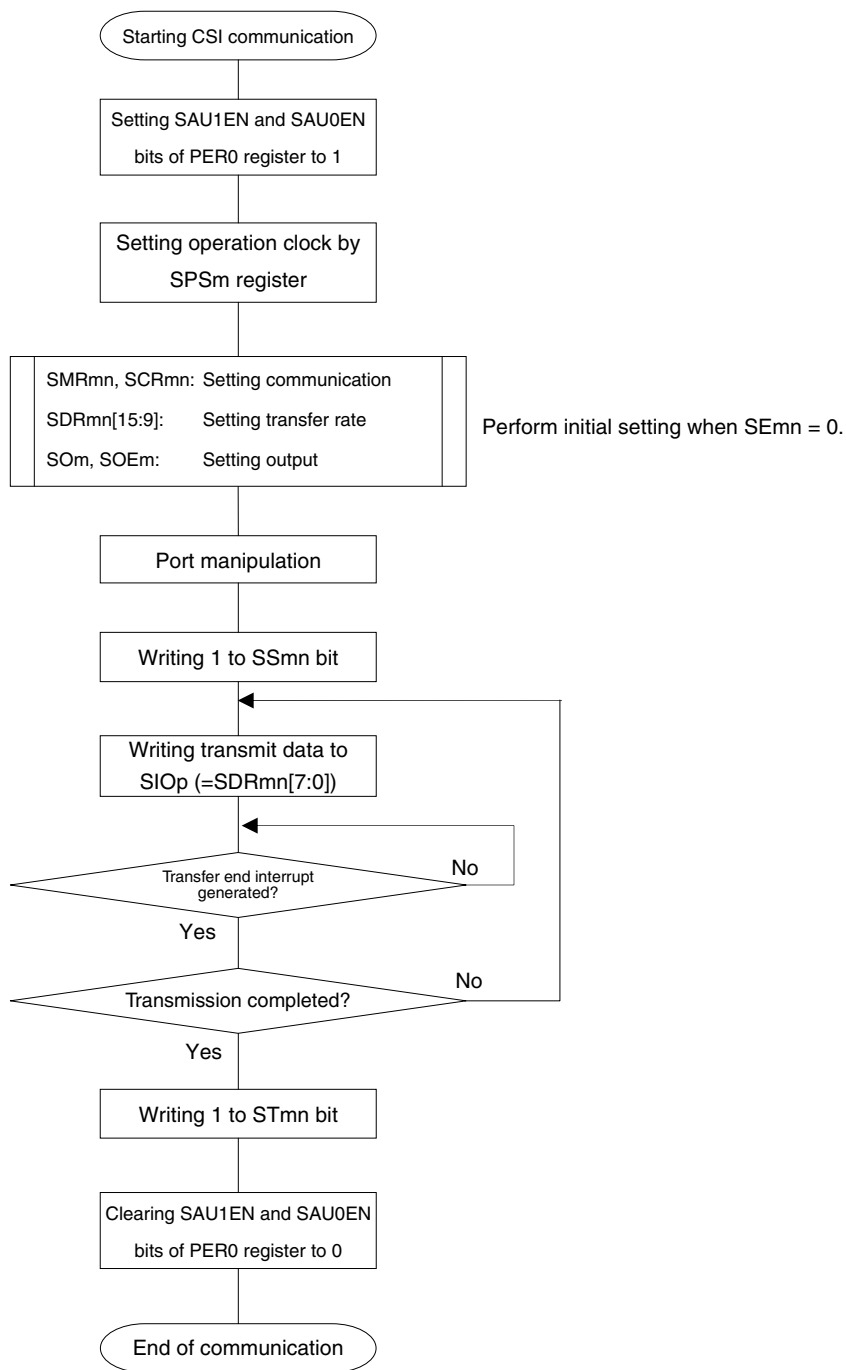
(3) Processing flow (in single-transmission mode)

Figure 13-28. Timing Chart of Master Transmission (in Single-Transmission Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

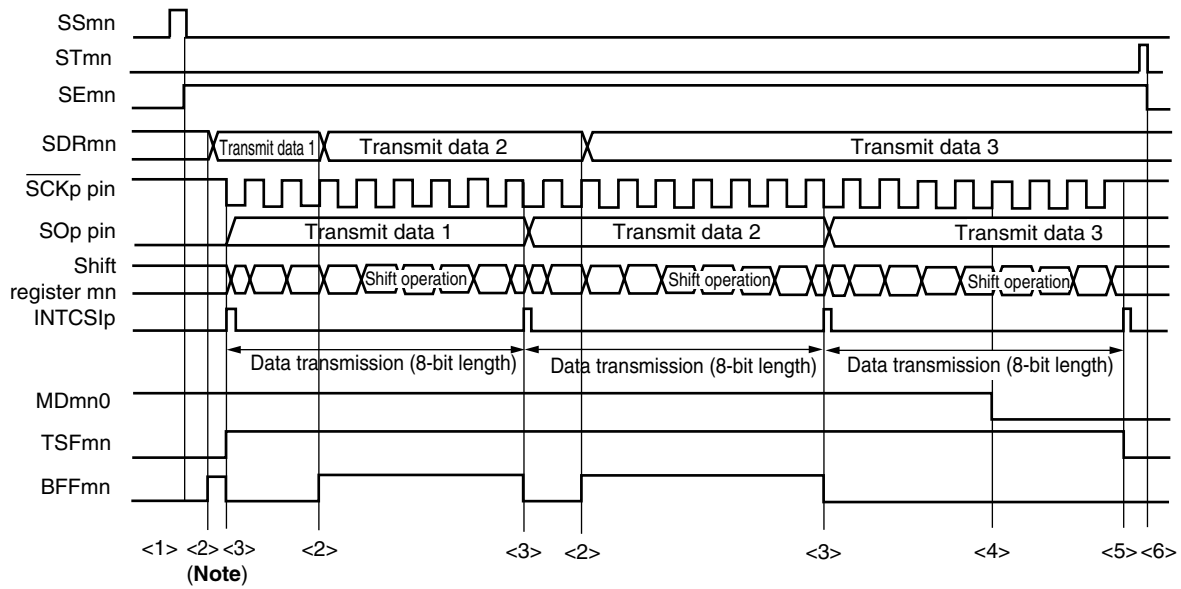
Figure 13-29. Flowchart of Master Transmission (in Single-Transmission Mode)



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 13-30. Timing Chart of Master Transmission (in Continuous Transmission Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)

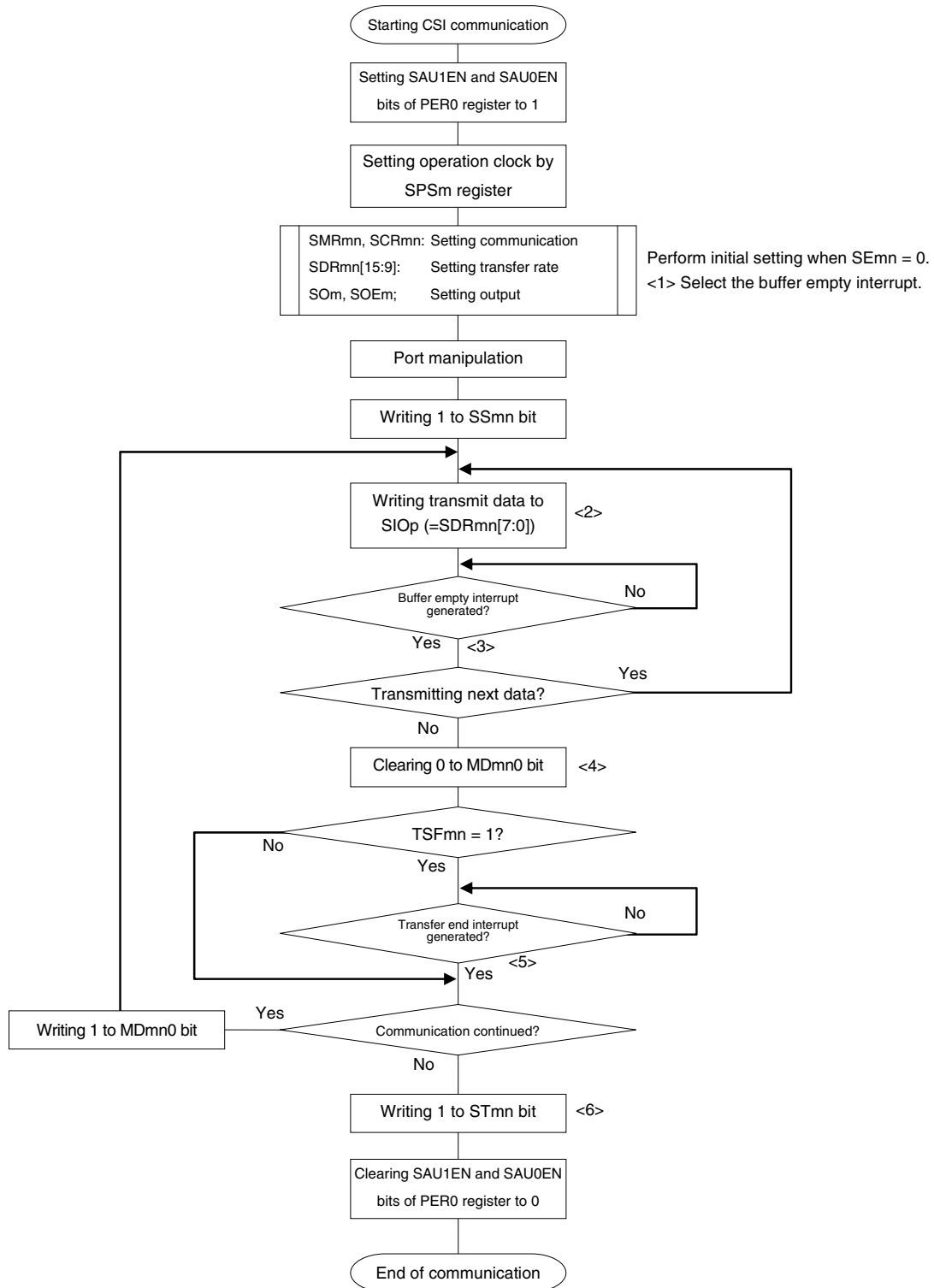


**Note** When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

**Caution** The MDmn0 bit can be rewritten even during operation.  
However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

Figure 13-31. Flowchart of Master Transmission (in Continuous Transmission Mode)



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

**Remark** <1> to <6> in the figure correspond to <1> to <6> in Figure 13-30 Timing Chart of Master Transmission (in Continuous Transmission Mode).

13.5.2 Master reception

Master reception is that the 78K0R/KH3 outputs a transfer clock and receives data from other device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK11, SI11	SCK20, SI20	SCK21, SI21
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.					
Error detection flag	Overrun error detection flag (OVFmn) only					
Transfer data length	7 or 8 bits					
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] <sup>Note</sup> $f_{CLK}$ : System clock frequency					
Data phase	Selectable by DAPmn bit <ul style="list-style-type: none"> <li>• DAPmn = 0: Data input starts from the start of the operation of the serial clock.</li> <li>• DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.</li> </ul>					
Clock phase	Selectable by CKPmn bit <ul style="list-style-type: none"> <li>• CKPmn = 0: Forward</li> <li>• CKPmn = 1: Reverse</li> </ul>					
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 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)** and **CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)**).

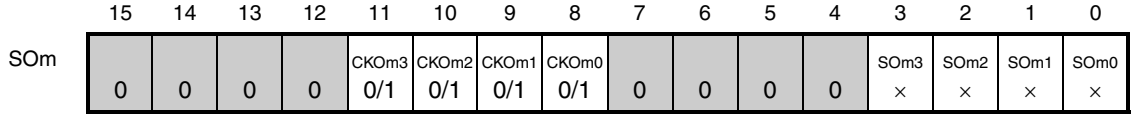
**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)



(1) Register setting

Figure 13-32. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

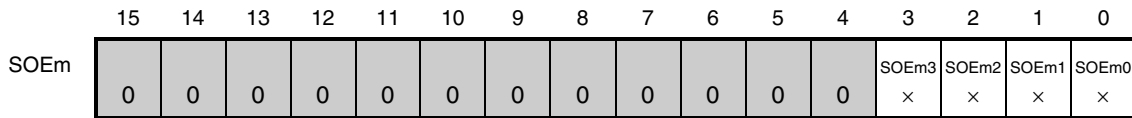
(a) Serial output register m (SOM) ... Sets only the bits of the target channel.



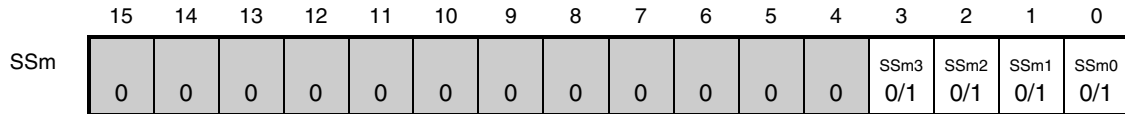
Communication starts when these bits are 1 if the data phase is forward (CKPmn = 0). If the phase is reversed (CKPmn = 1), communication starts when these bits are 0.

<R>

(b) Serial output enable register m (SOEm) ... The register that not used in this mode.

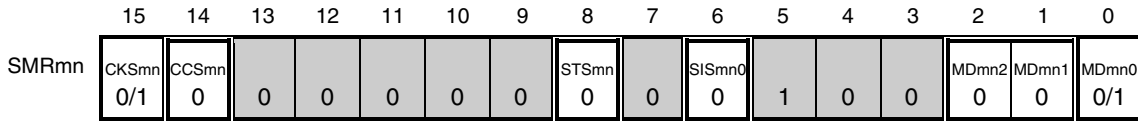


(c) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.



<R>

(d) Serial mode register mn (SMRmn)

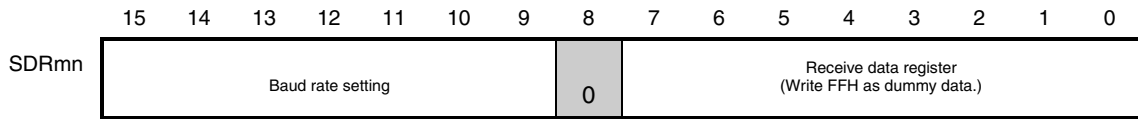


Interrupt sources of channel n  
 0: Transfer end interrupt  
 1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)



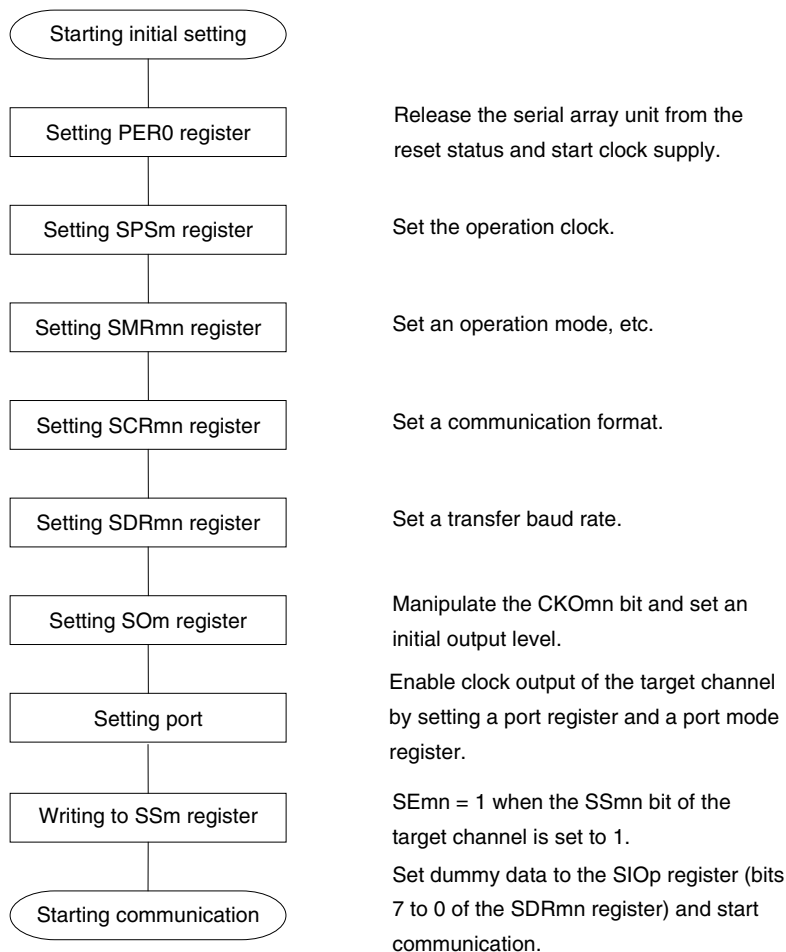
(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)  
: Setting is fixed in the CSI master reception mode, : Setting disabled (set to the initial value)  
 ×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)  
 0/1: Set to 0 or 1 depending on the usage of the user

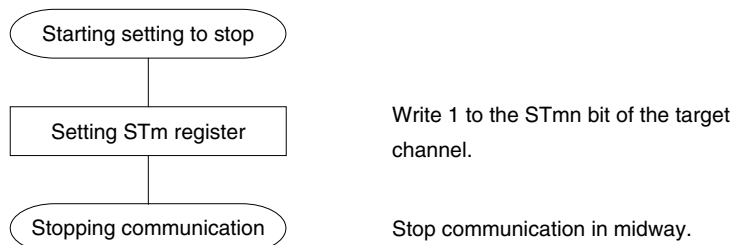
(2) Operation procedure

Figure 13-33. Initial Setting Procedure for Master Reception



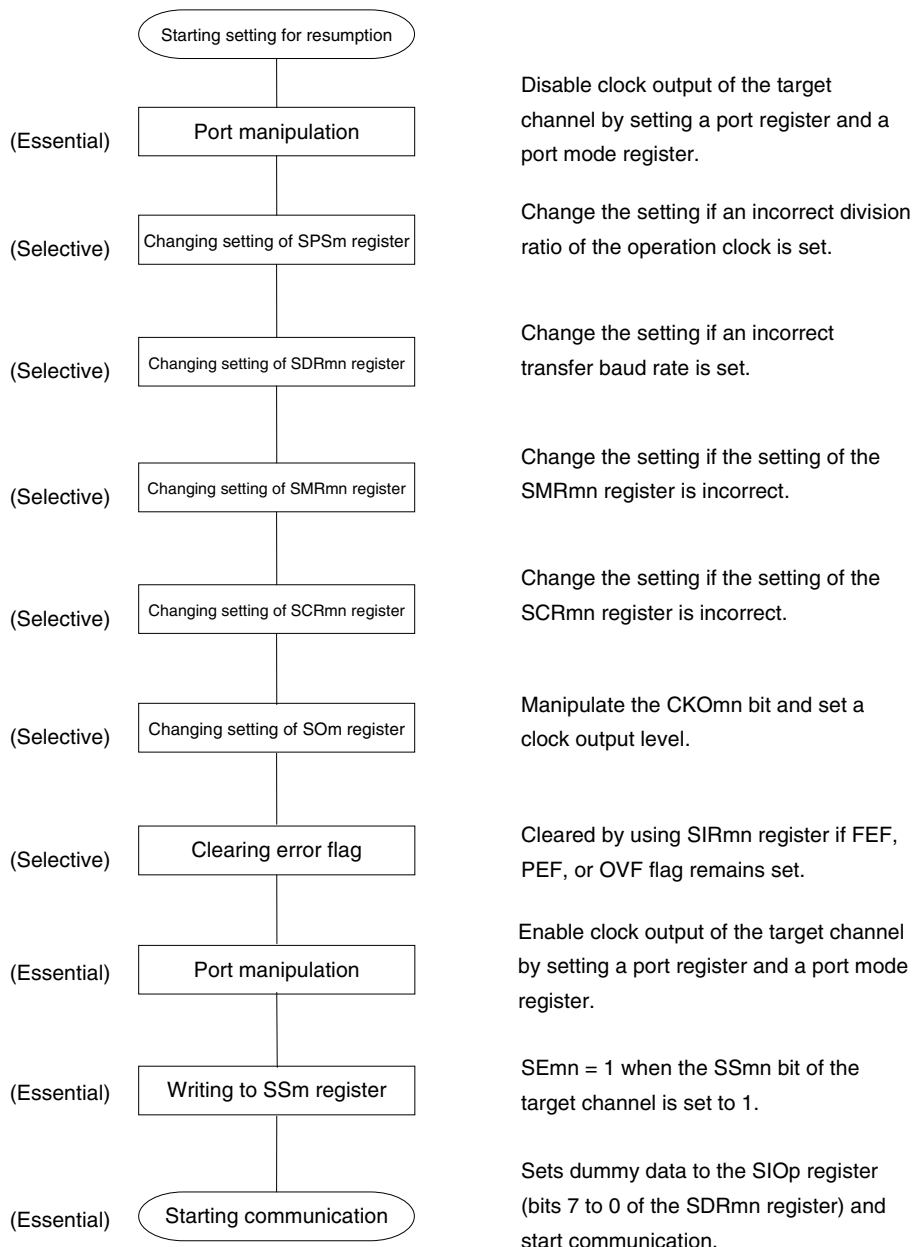
**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 13-34. Procedure for Stopping Master Reception



**Remark** Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOM register (see **Figure 13-35 Procedure for Resuming Master Reception**).

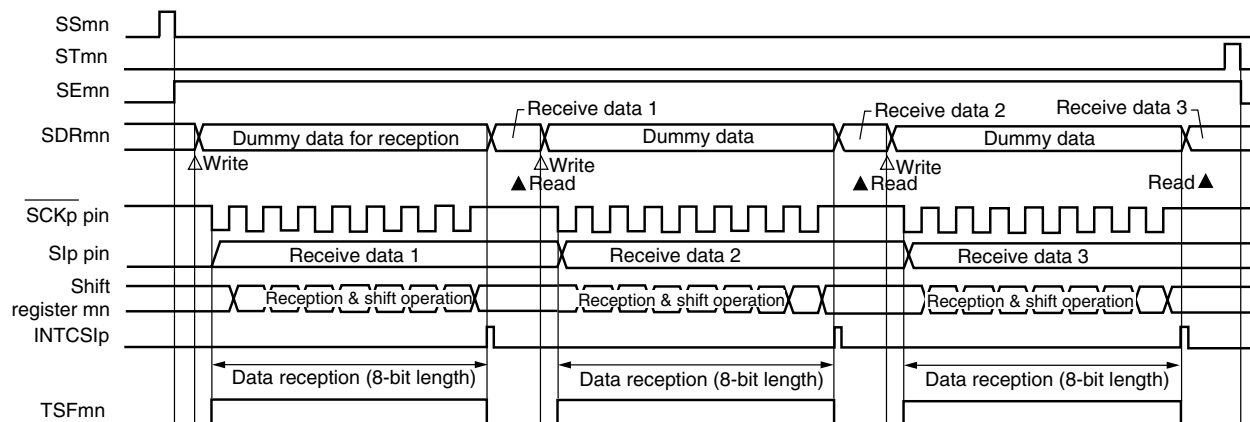
**Figure 13-35. Procedure for Resuming Master Reception**



<R>

(3) Processing flow (in single-reception mode)

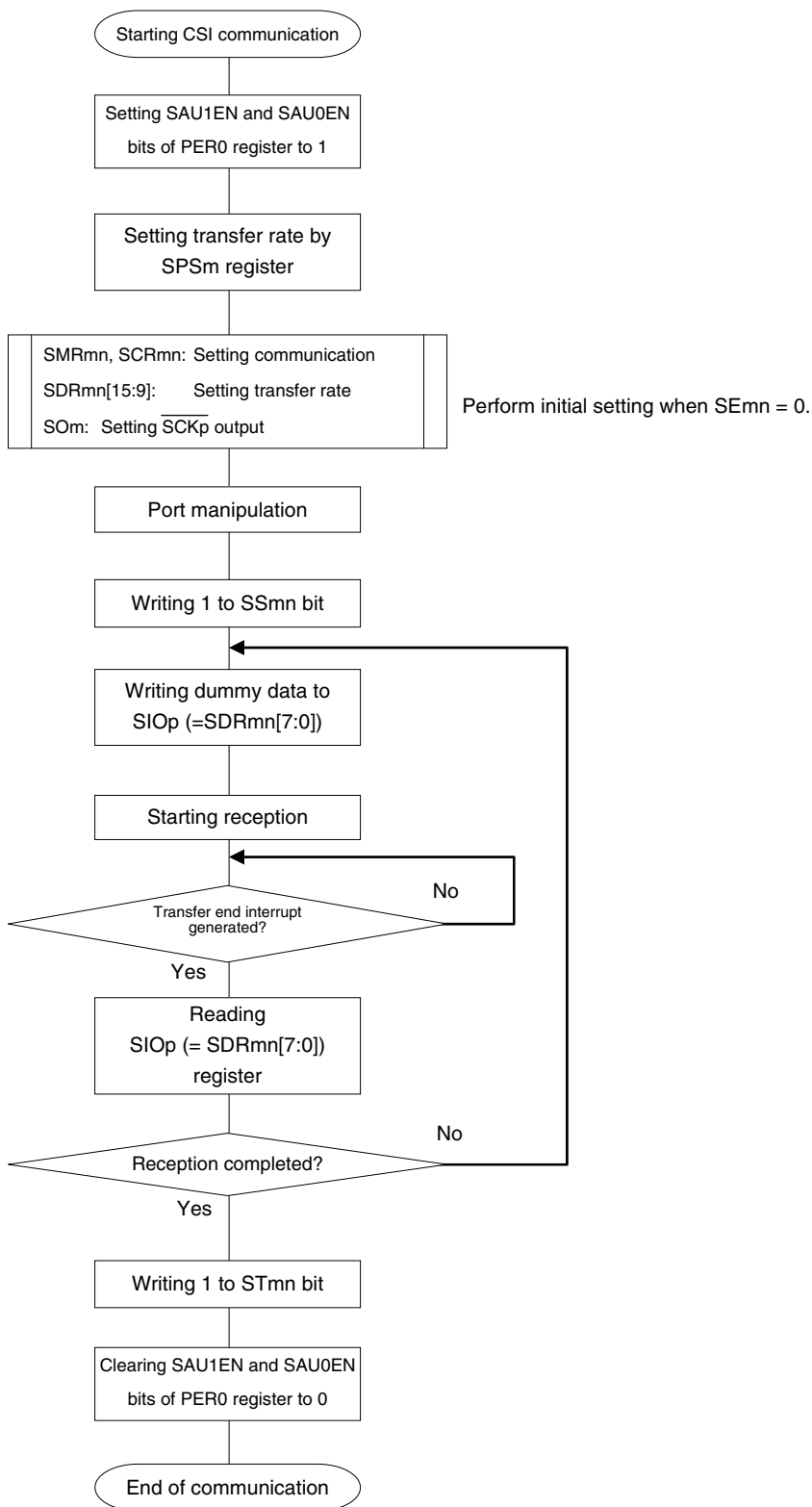
Figure 13-36. Timing Chart of Master Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

Figure 13-37. Flowchart of Master Reception (in Single-Reception Mode)

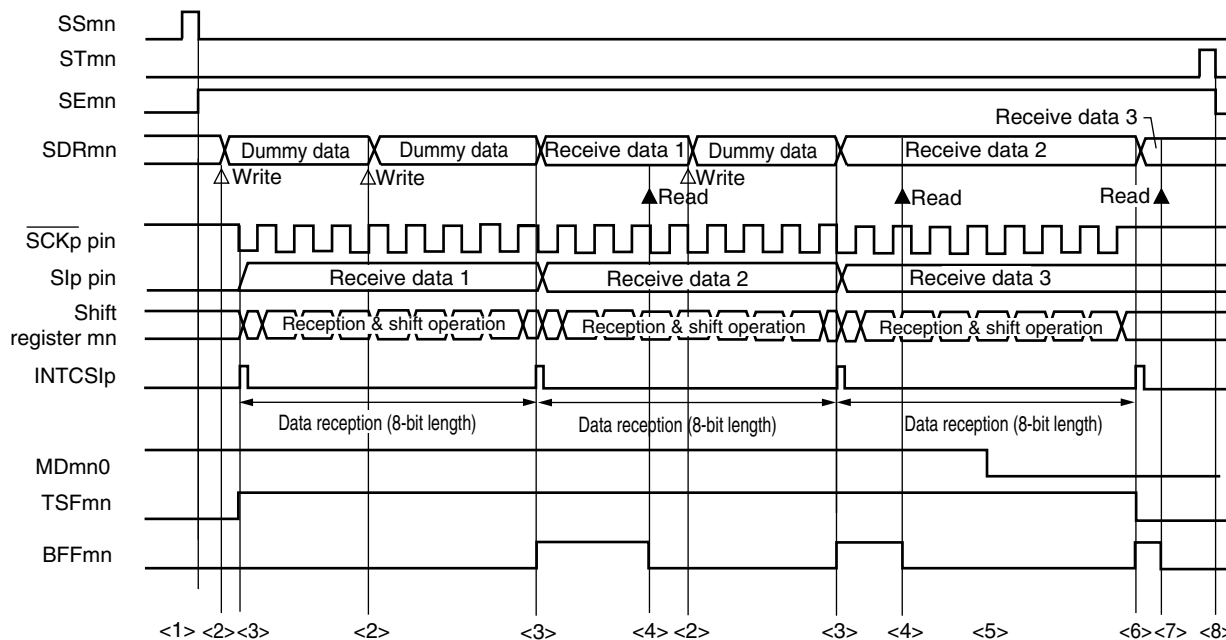
<R>



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous reception mode)

<R> Figure 13-38. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



**Caution** The MDmn0 bit can be rewritten even during operation.

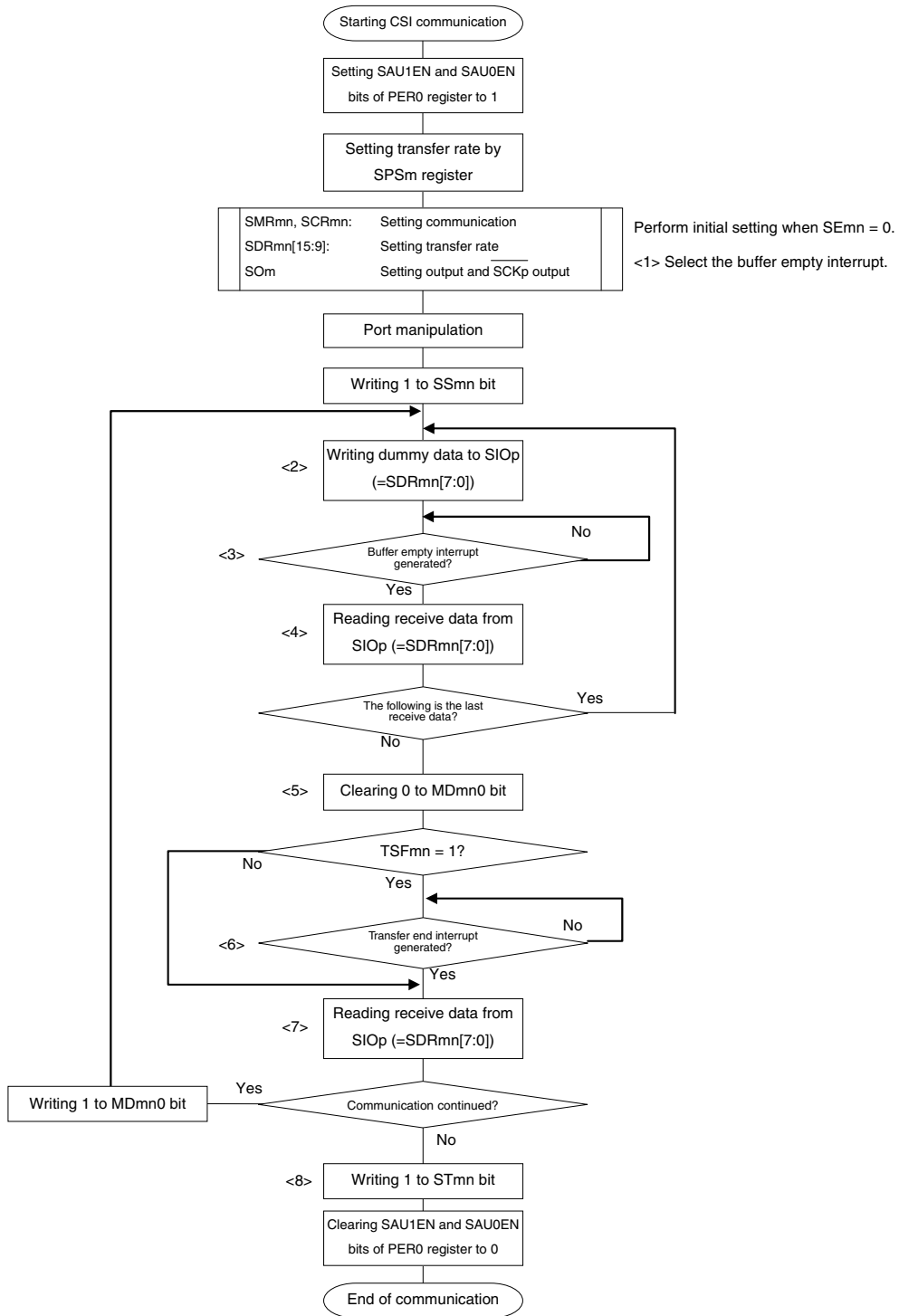
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

**Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in **Figure 13-39 Flowchart of Master Reception (in Continuous Reception Mode)**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

<R>

Figure 13-39. Flowchart of Master Reception (in Continuous Reception Mode)



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

**Remark** <1> to <8> in the figure correspond to <1> to <8> in Figure 13-38 Timing Chart of Master Reception (in Continuous Reception Mode).

**13.5.3 Master transmission/reception**

Master transmission/reception is that the 78K0R/KH3 outputs a transfer clock and transmits/receives data to/from other device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK11, SI11, SO11	SCK20, SI20, SO20	SCK21, SI21, SO21
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.					
Error detection flag	Overrun error detection flag (OVFmn) only					
Transfer data length	7 or 8 bits					
Transfer rate	Max. $f_{CLK}/4$ [Hz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [Hz] <sup>Note</sup> $f_{CLK}$ : System clock frequency					
Data phase	Selectable by DAPmn bit <ul style="list-style-type: none"> <li>• DAPmn = 0: Data output starts at the start of the operation of the serial clock.</li> <li>• DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.</li> </ul>					
Clock phase	Selectable by CKPmn bit <ul style="list-style-type: none"> <li>• CKPmn = 0: Forward</li> <li>• CKPmn = 1: Reverse</li> </ul>					
Data direction	MSB or LSB first					

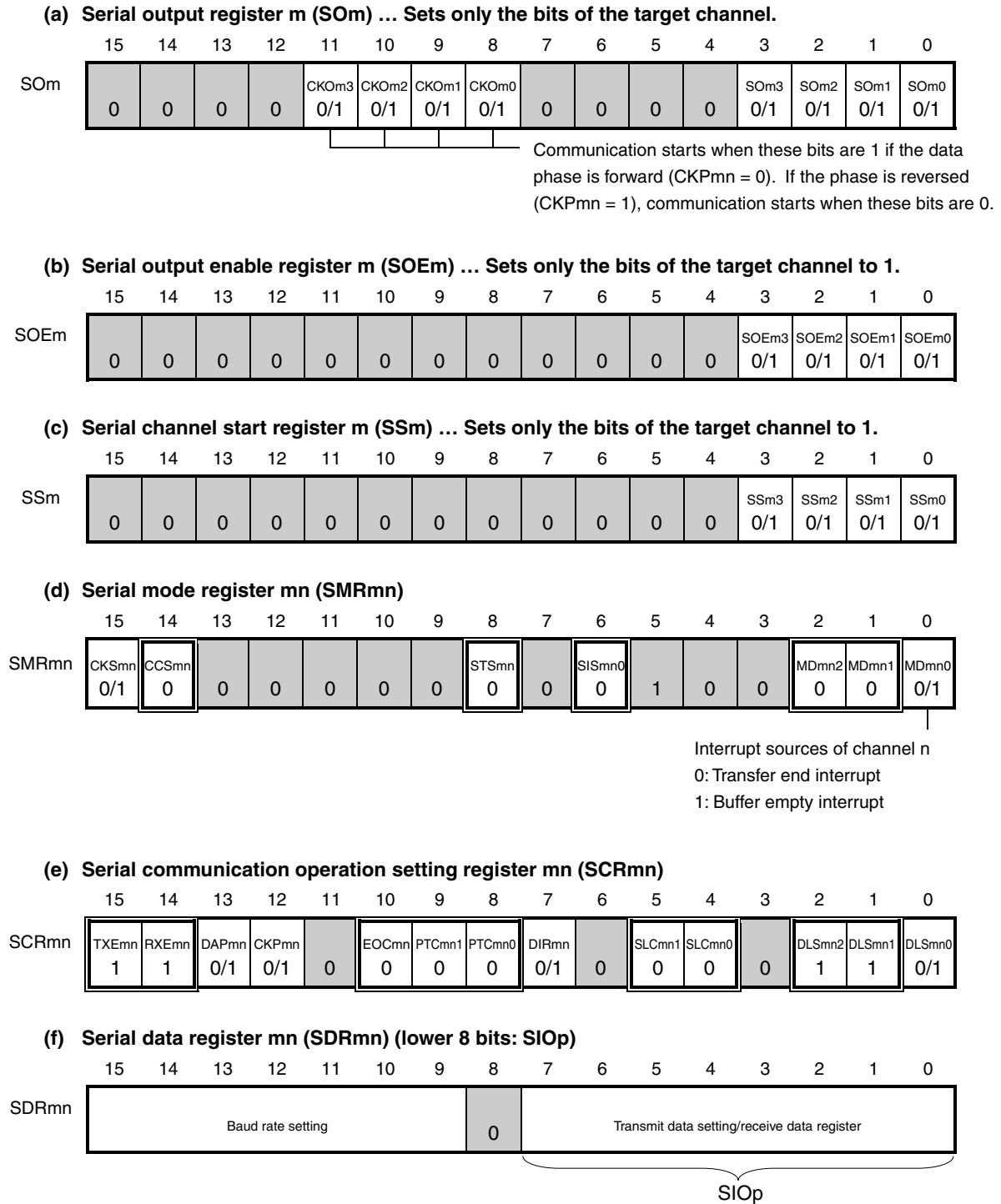
**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)** and **CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)**).

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)



(1) Register setting

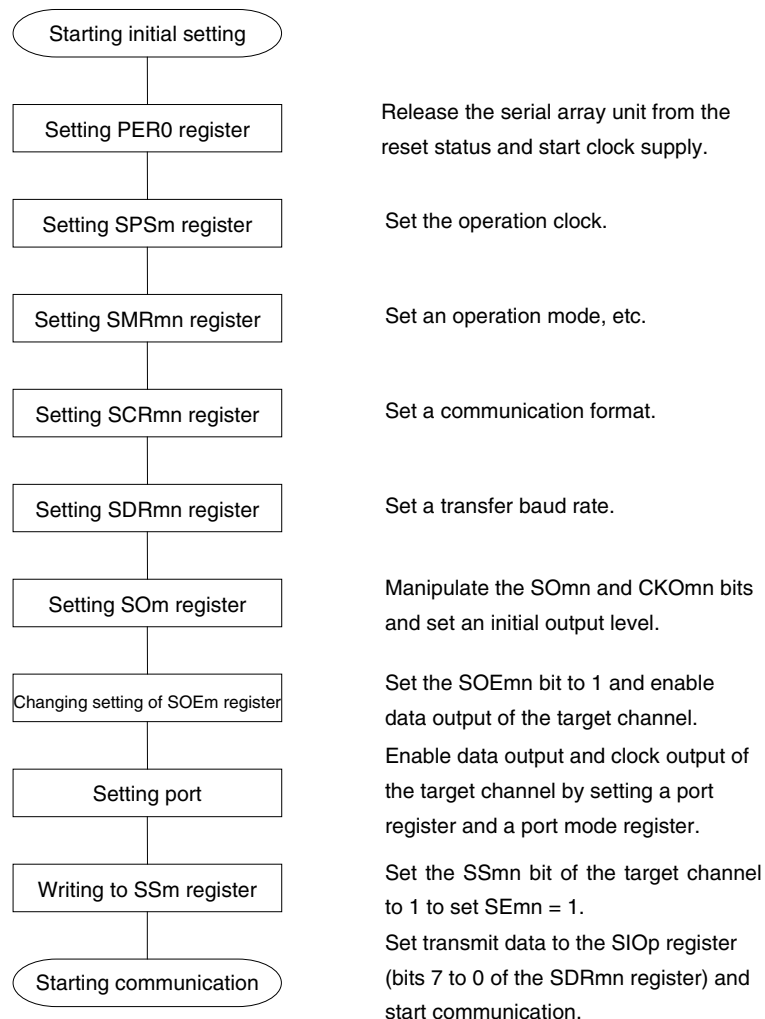
Figure 13-40. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)  
: Setting is fixed in the CSI master transmission/reception mode, : Setting disabled (set to the initial value)  
 0/1: Set to 0 or 1 depending on the usage of the user

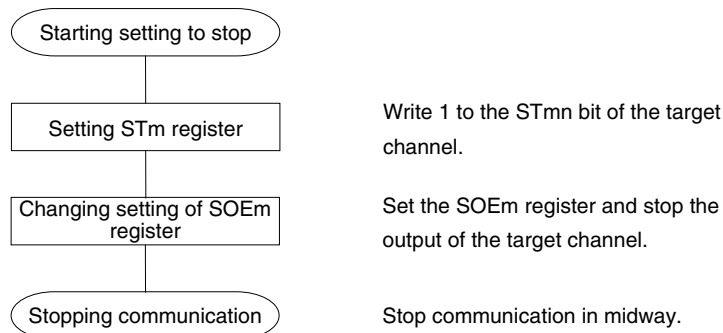
(2) Operation procedure

Figure 13-41. Initial Setting Procedure for Master Transmission/Reception



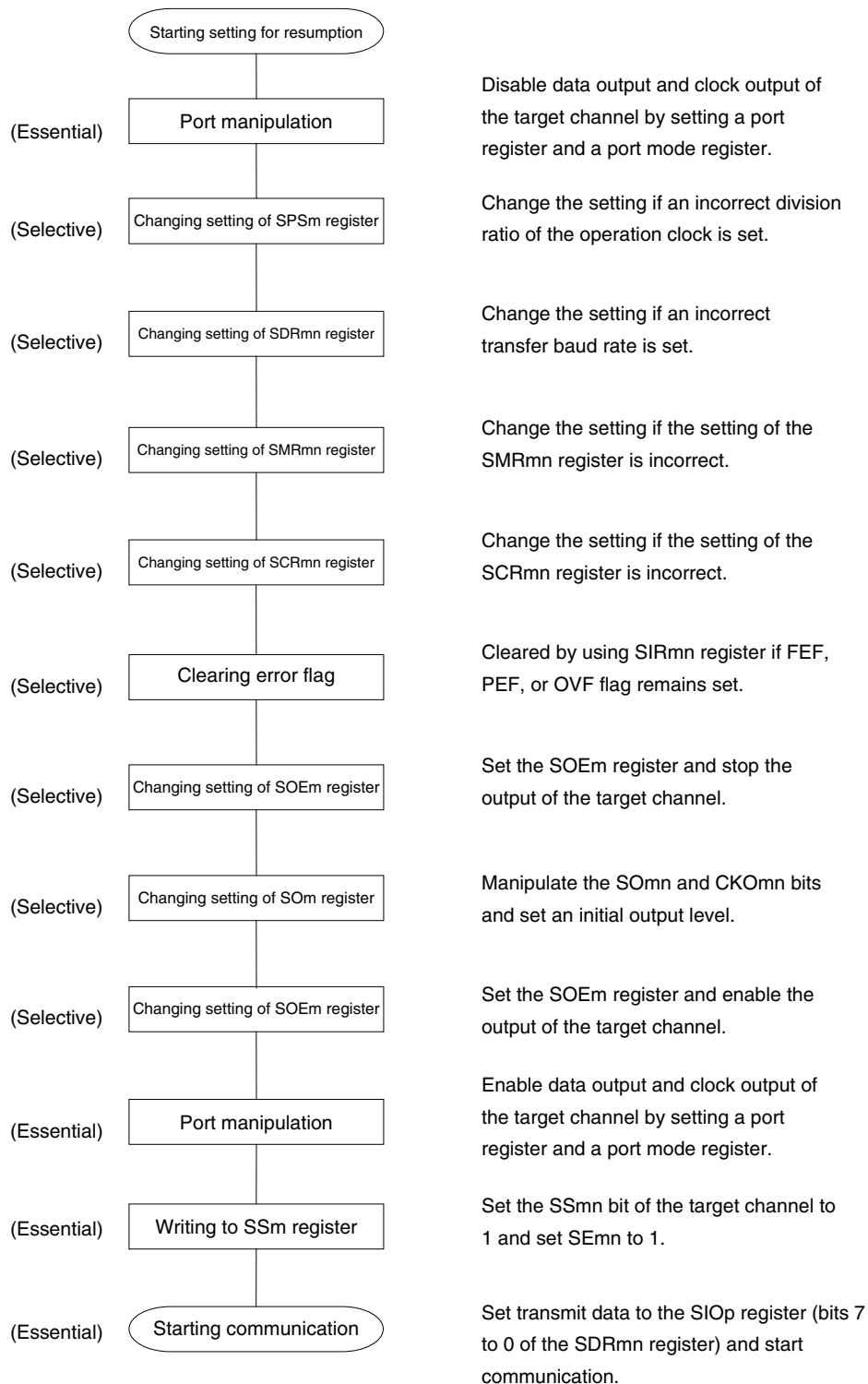
**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 13-42. Procedure for Stopping Master Transmission/Reception



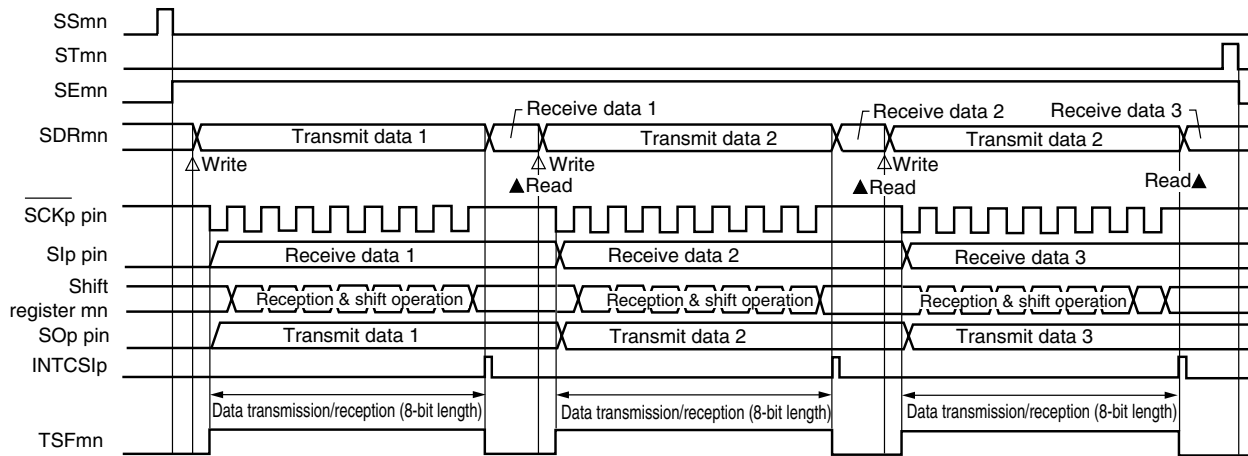
**Remark** Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOM register (see **Figure 13-43 Procedure for Resuming Master Transmission/Reception**).

Figure 13-43. Procedure for Resuming Master Transmission/Reception



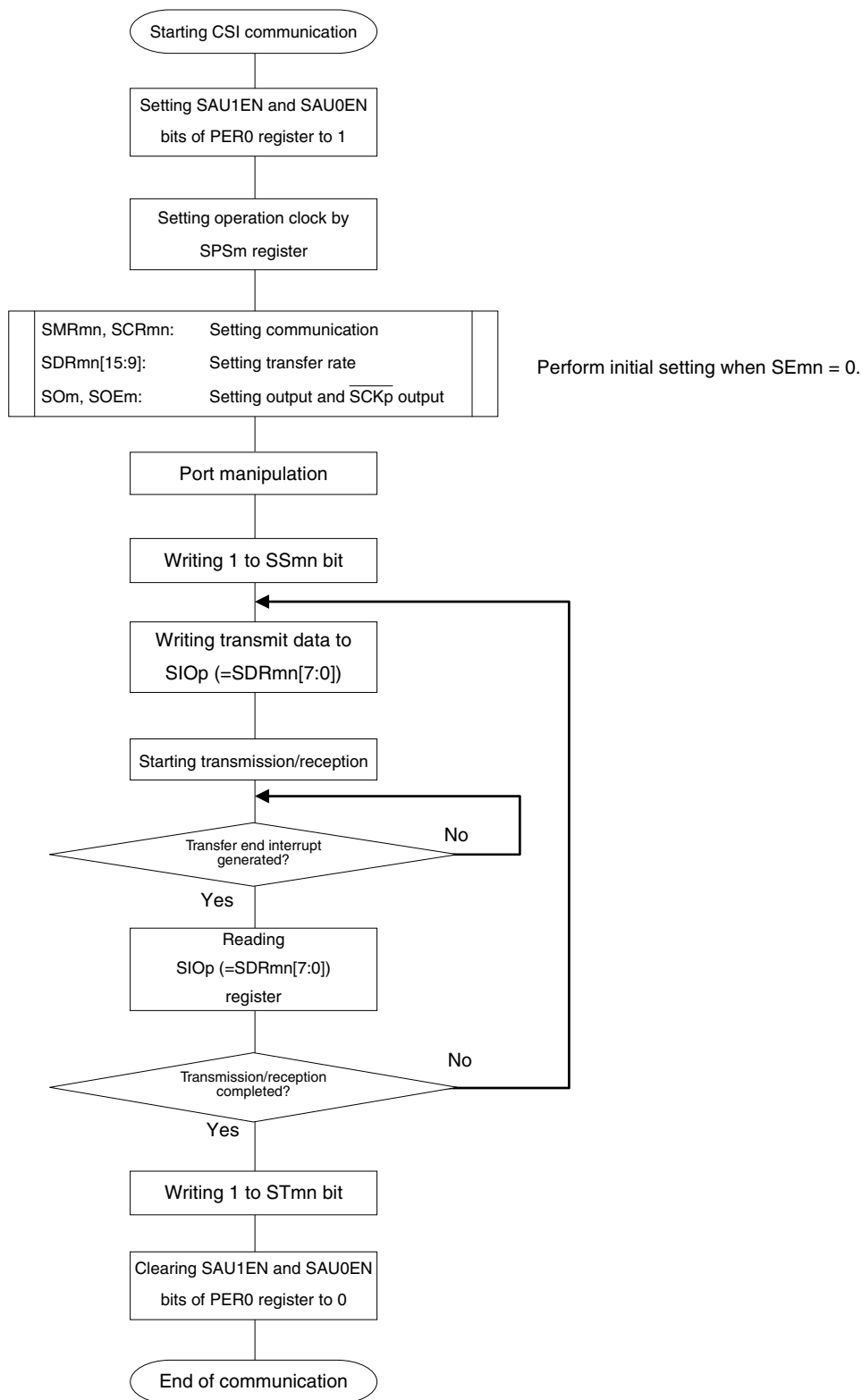
(3) Processing flow (in single-transmission/reception mode)

Figure 13-44. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

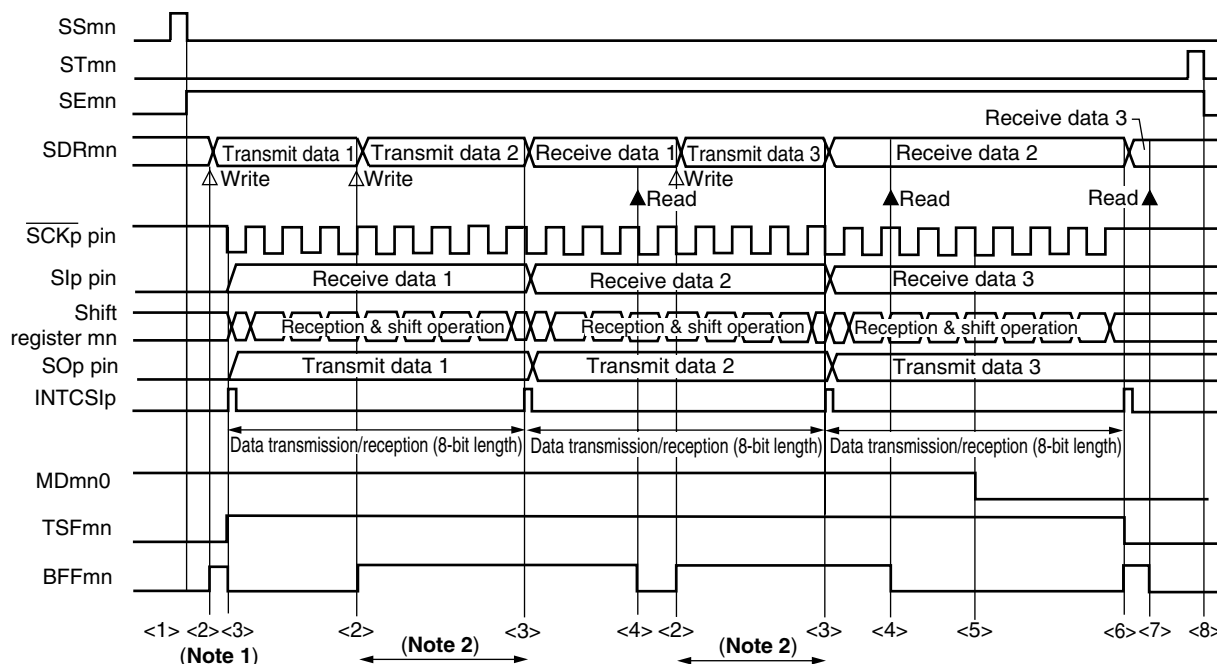
Figure 13-45. Flowchart of Master Transmission/Reception (in Single-Transmission/Reception Mode)



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission/reception mode)

Figure 13-46. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)

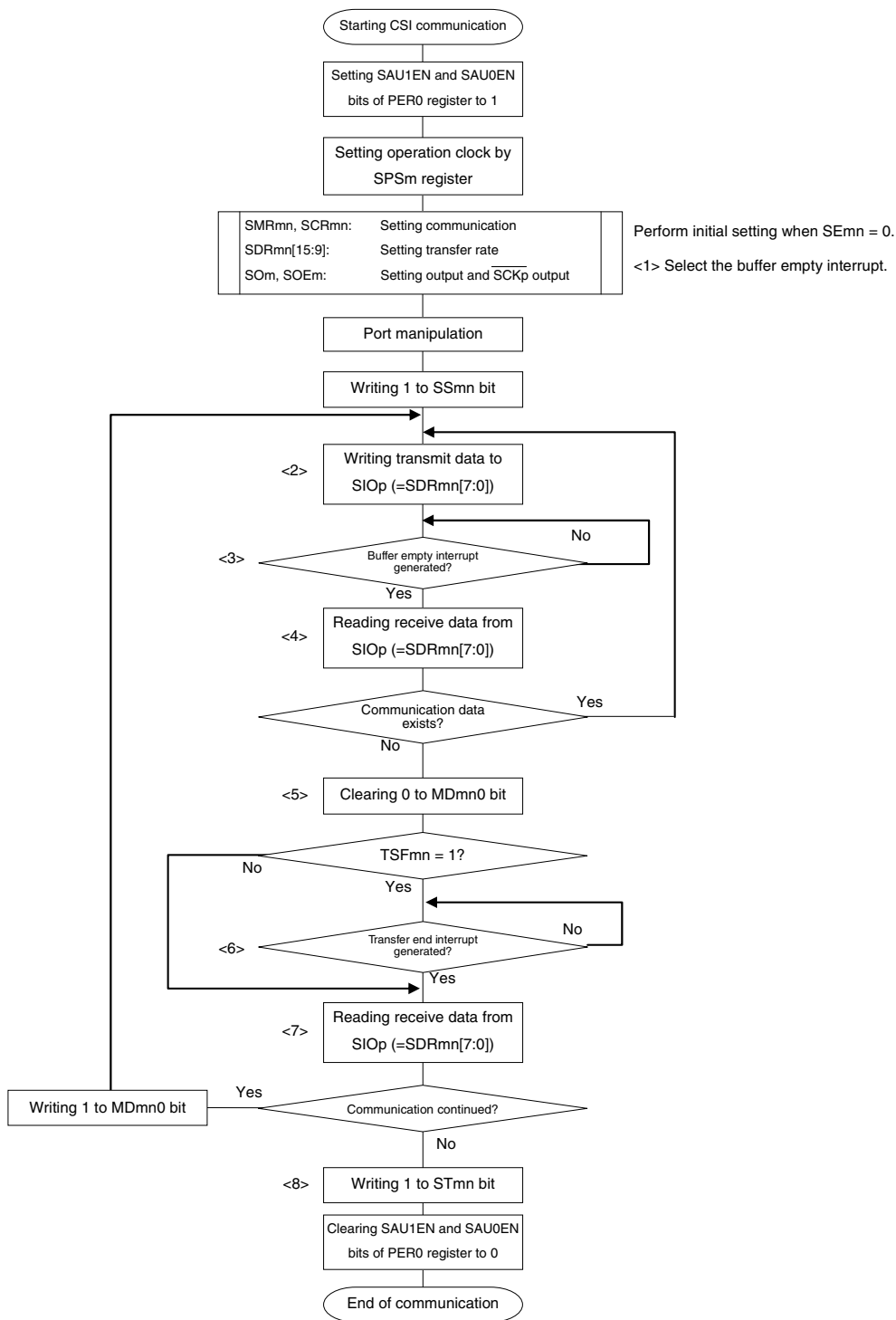


- Notes**
1. When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.
  2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

**Caution** The MDmn0 bit 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 13-47 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.
  2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

Figure 13-47. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

**Remark** <1> to <8> in the figure correspond to <1> to <8> in Figure 13-46 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode).

### 13.5.4 Slave transmission

Slave transmission is that the 78K0R/KH3 transmits data to another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	$\overline{\text{SCK00}}$ , $\overline{\text{SO00}}$	$\overline{\text{SCK01}}$ , $\overline{\text{SO01}}$	$\overline{\text{SCK10}}$ , $\overline{\text{SO10}}$	$\overline{\text{SCK11}}$ , $\overline{\text{SO11}}$	$\overline{\text{SCK20}}$ , $\overline{\text{SO20}}$	$\overline{\text{SCK21}}$ , $\overline{\text{SO21}}$
Interrupt	$\overline{\text{INTCSI00}}$	$\overline{\text{INTCSI01}}$	$\overline{\text{INTCSI10}}$	$\overline{\text{INTCSI11}}$	$\overline{\text{INTCSI20}}$	$\overline{\text{INTCSI21}}$
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.					
Error detection flag	Overrun error detection flag (OVFmn) only					
Transfer data length	7 or 8 bits					
Transfer rate	Max. $f_{\text{MCK}}/6$ [Hz] <sup>Notes 1, 2</sup>					
Data phase	Selectable by DAPmn bit <ul style="list-style-type: none"> <li>• DAPmn = 0: Data output starts from the start of the operation of the serial clock.</li> <li>• DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.</li> </ul>					
Clock phase	Selectable by CKPmn bit <ul style="list-style-type: none"> <li>• CKPmn = 0: Forward</li> <li>• CKPmn = 1: Reverse</li> </ul>					
Data direction	MSB or LSB first					

**Notes 1.** Because the external serial clock input to pins  $\overline{\text{SCK00}}$ ,  $\overline{\text{SCK01}}$ ,  $\overline{\text{SCK10}}$ ,  $\overline{\text{SCK11}}$ ,  $\overline{\text{SCK20}}$ , and  $\overline{\text{SCK21}}$  is sampled internally and used, the fastest transfer rate is  $f_{\text{MCK}}/6$  [Hz].

**2.** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)** and **CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)**).

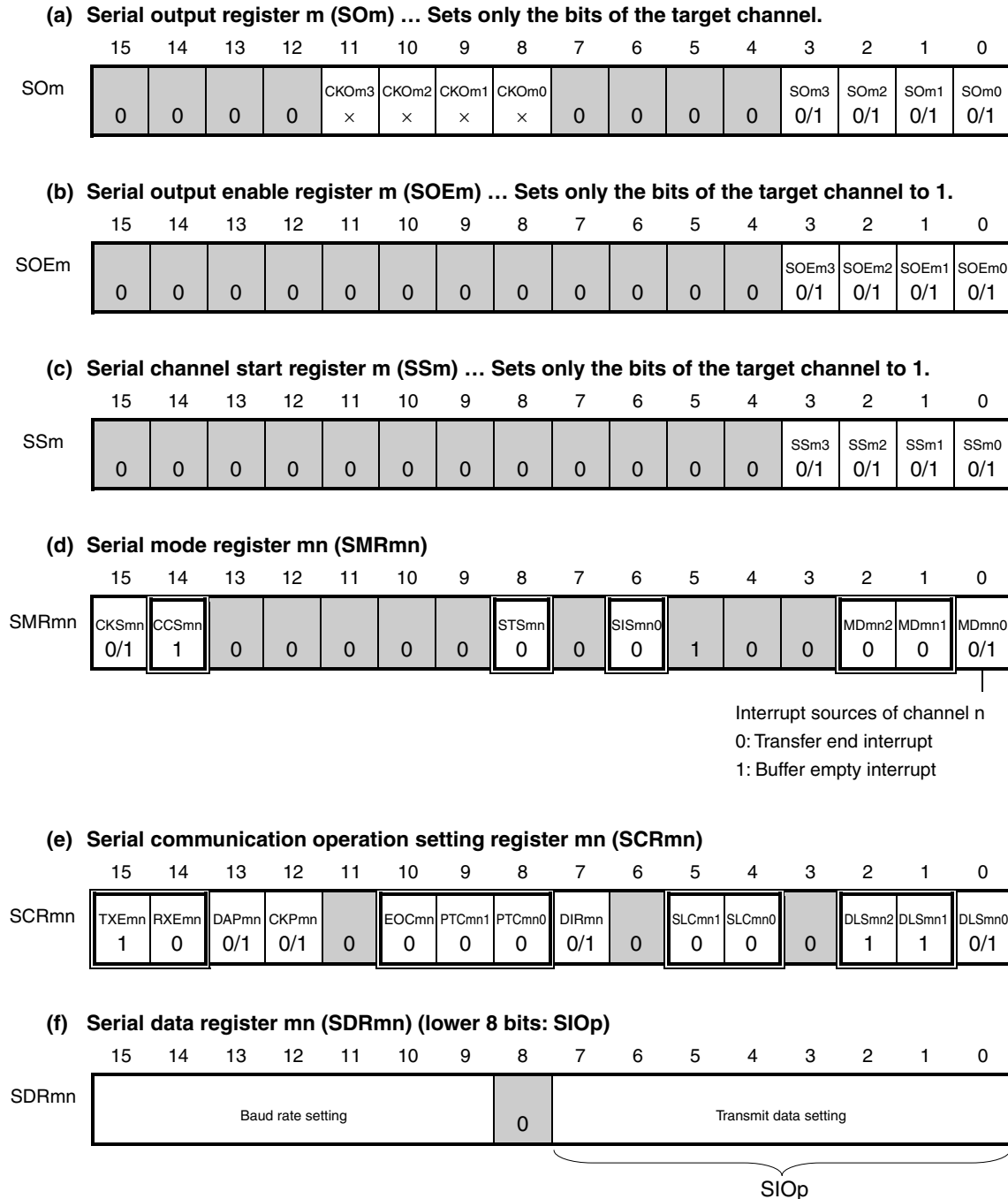
**Remarks 1.**  $f_{\text{MCK}}$ : Operation clock (MCK) frequency of target channel

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)



(1) Register setting

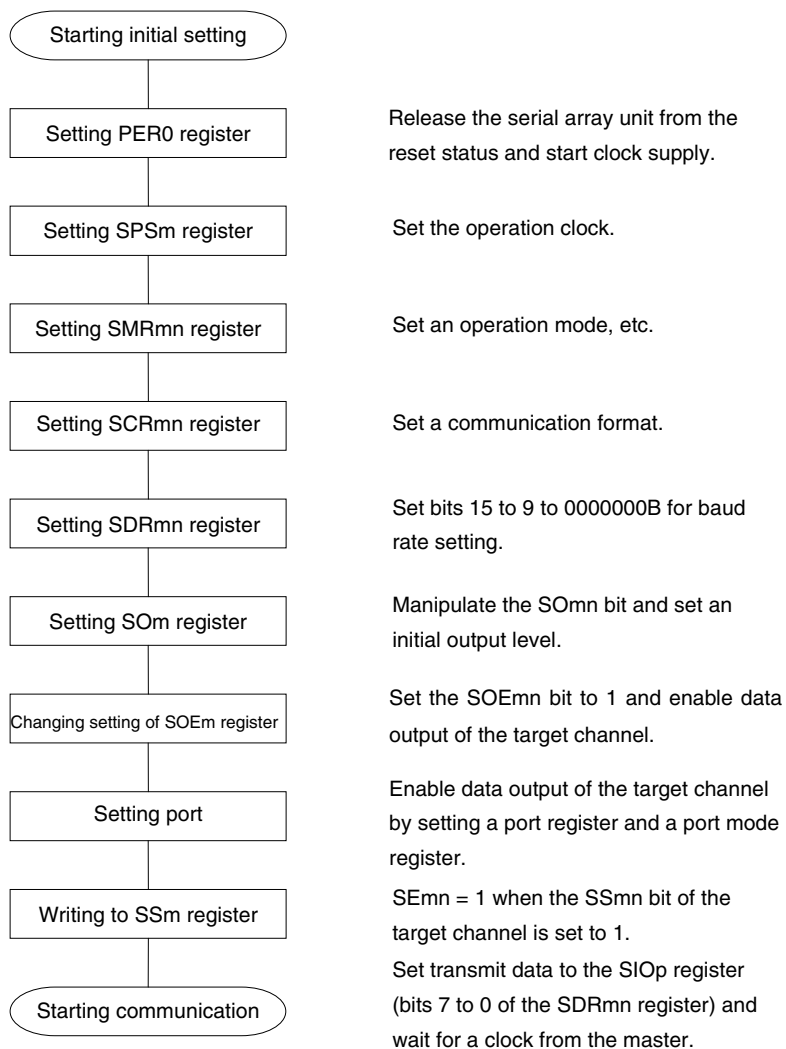
Figure 13-48. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)



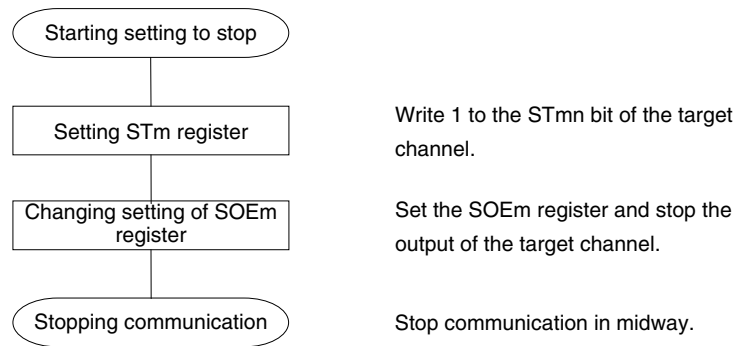
**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)  
: Setting is fixed in the CSI slave transmission mode, : Setting disabled (set to the initial value)  
 x: Bit that cannot be used in this mode (set to the initial value when not used in any mode)  
 0/1: Set to 0 or 1 depending on the usage of the user

## (2) Operation procedure

Figure 13-49. Initial Setting Procedure for Slave Transmission

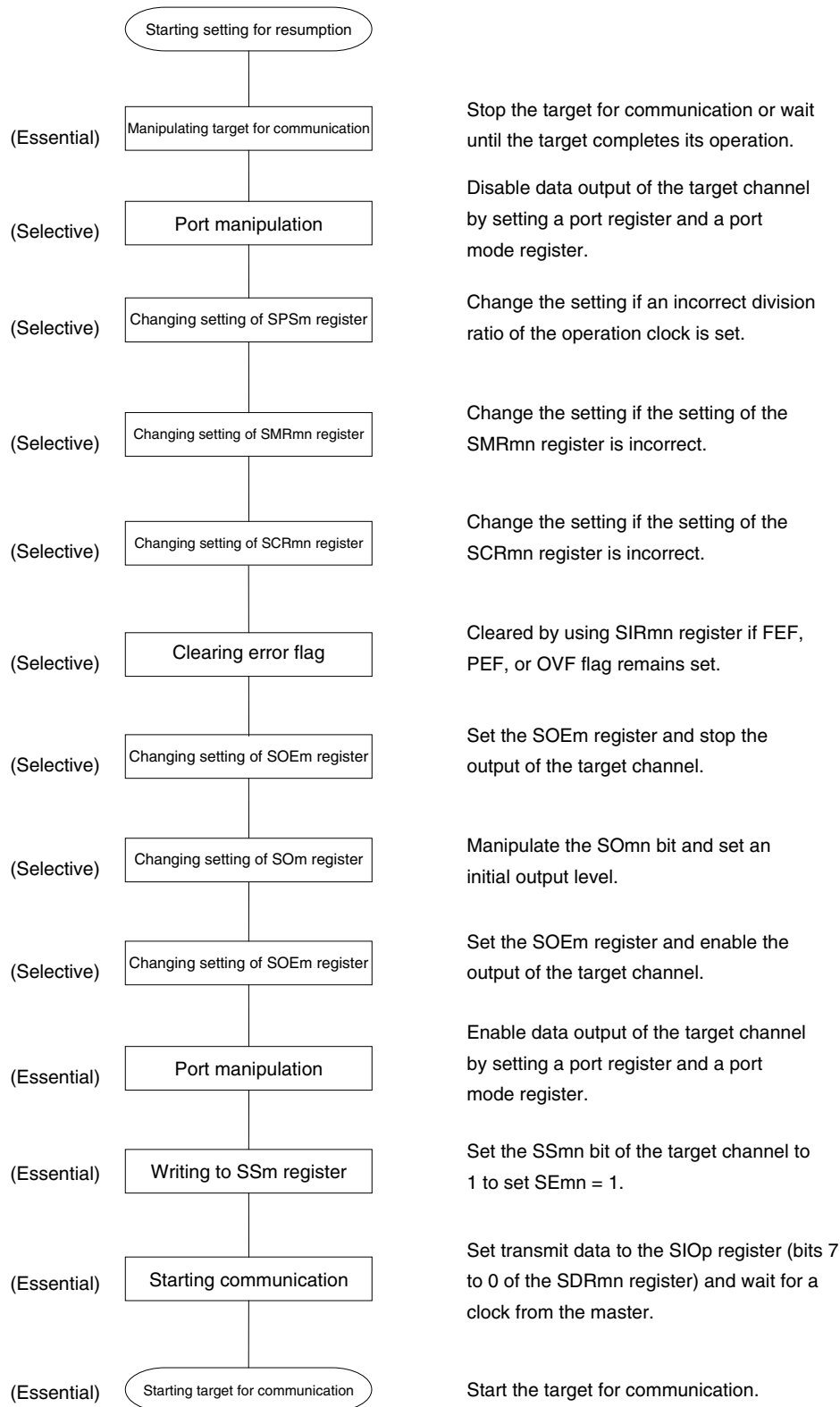


**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

**Figure 13-50. Procedure for Stopping Slave Transmission**

**Remark** Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 13-51 Procedure for Resuming Slave Transmission**).

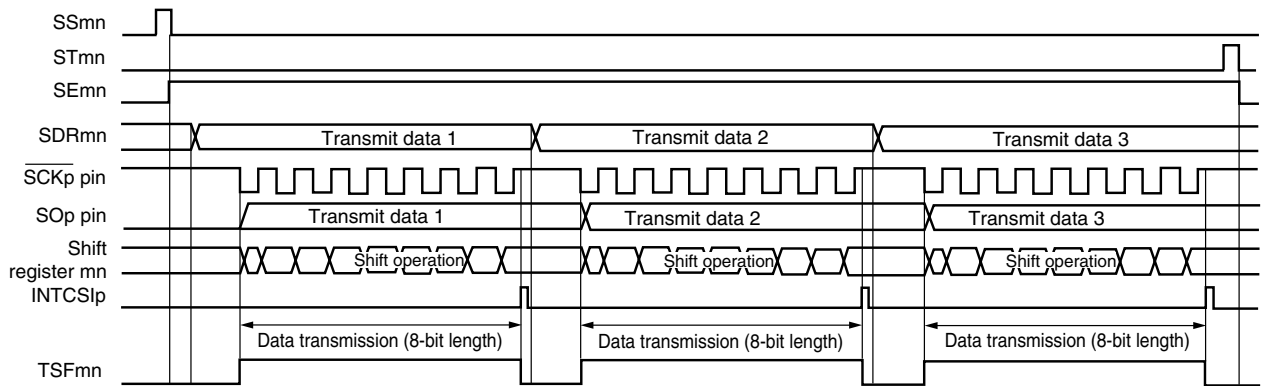
Figure 13-51. Procedure for Resuming Slave Transmission



<R>

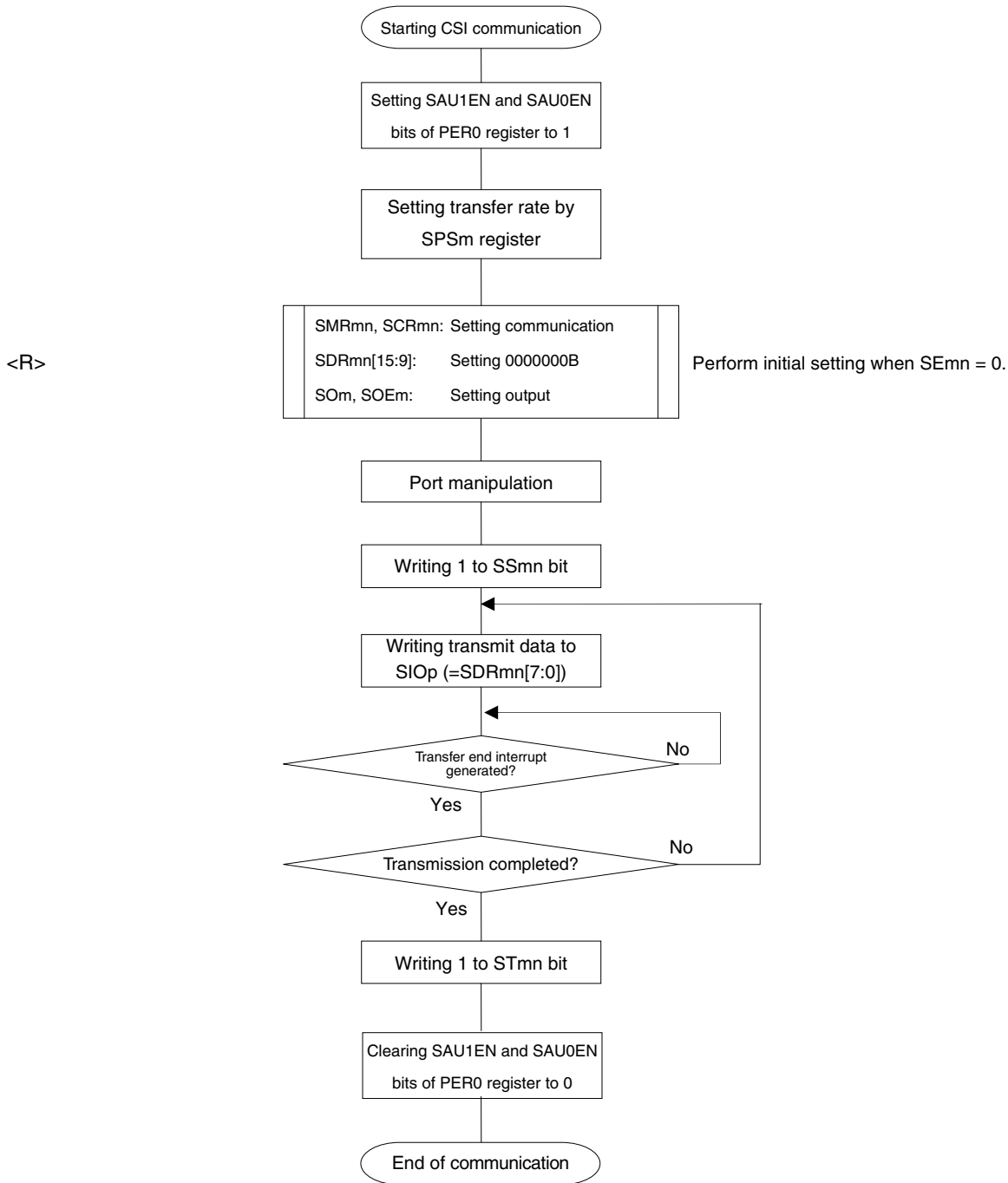
## (3) Processing flow (in single-transmission mode)

**Figure 13-52. Timing Chart of Slave Transmission (in Single-Transmission Mode)**  
 (Type 1: DAPmn = 0, CKPmn = 0)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

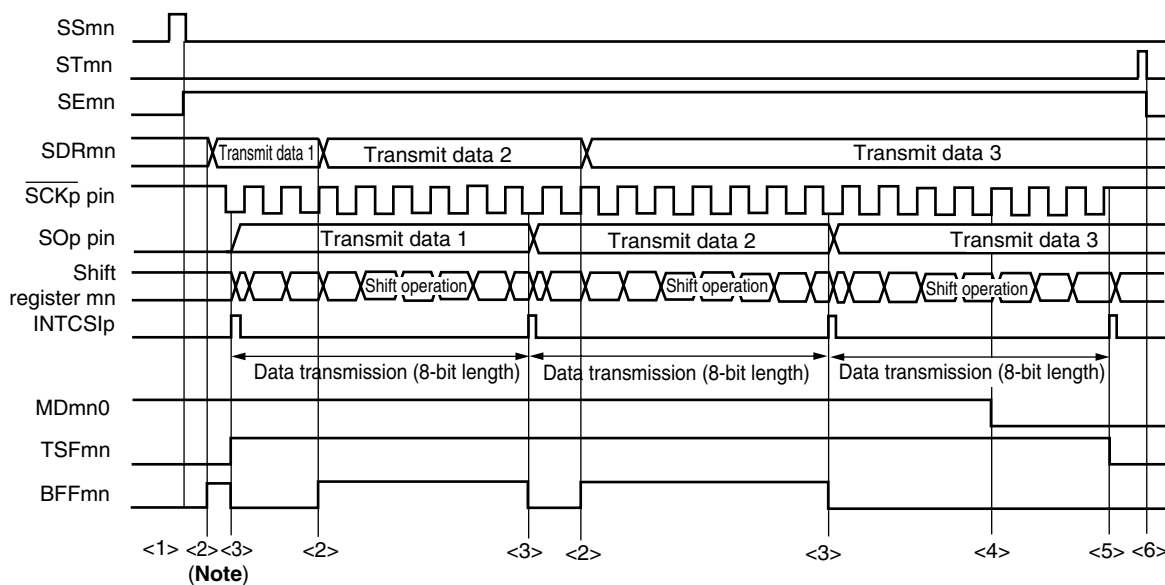
Figure 13-53. Flowchart of Slave Transmission (in Single-Transmission Mode)



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 13-54. Timing Chart of Slave Transmission (in Continuous Transmission Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)

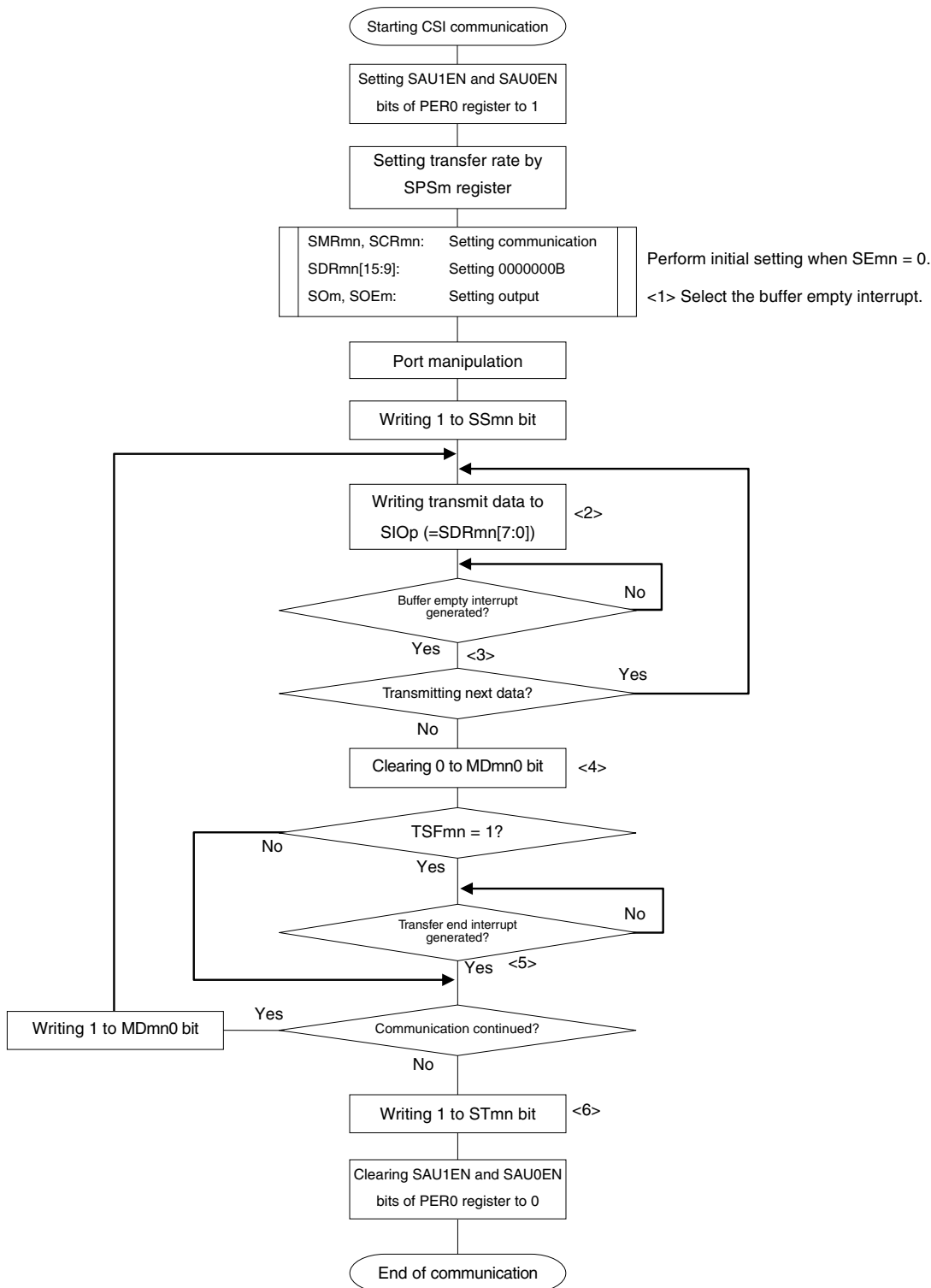


**Note** When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

**Caution** The MDmn0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Figure 13-55. Flowchart of Slave Transmission (in Continuous Transmission Mode)

<R>



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

**Remark** <1> to <6> in the figure correspond to <1> to <6> in Figure 13-54 Timing Chart of Slave Transmission (in Continuous Transmission Mode).



### 13.5.5 Slave reception

Slave reception is that the 78K0R/KH3 receives data from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	$\overline{\text{SCK00}}$ , SI00	$\overline{\text{SCK01}}$ , SI01	$\overline{\text{SCK10}}$ , SI10	$\overline{\text{SCK11}}$ , SI11	$\overline{\text{SCK20}}$ , SI20	$\overline{\text{SCK21}}$ , SI21
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)					
Error detection flag	Overrun error detection flag (OVFmn) only					
Transfer data length	7 or 8 bits					
Transfer rate	Max. $f_{\text{MCK}}/6$ [Hz] <sup>Notes 1, 2</sup>					
Data phase	Selectable by DAPmn bit <ul style="list-style-type: none"> <li>• DAPmn = 0: Data input starts from the start of the operation of the serial clock.</li> <li>• DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.</li> </ul>					
Clock phase	Selectable by CKPmn bit <ul style="list-style-type: none"> <li>• CKPmn = 0: Forward</li> <li>• CKPmn = 1: Reverse</li> </ul>					
Data direction	MSB or LSB first					

**Notes 1.** Because the external serial clock input to pins  $\overline{\text{SCK00}}$ ,  $\overline{\text{SCK01}}$ ,  $\overline{\text{SCK10}}$ ,  $\overline{\text{SCK11}}$ ,  $\overline{\text{SCK20}}$ , and  $\overline{\text{SCK21}}$  is sampled internally and used, the fastest transfer rate is  $f_{\text{MCK}}/6$  [Hz].

**2.** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)** and **CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)**).

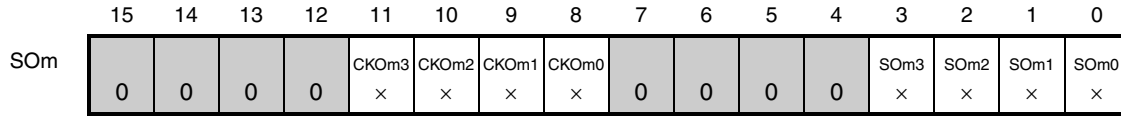
**Remarks 1.**  $f_{\text{MCK}}$ : Operation clock (MCK) frequency of target channel

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

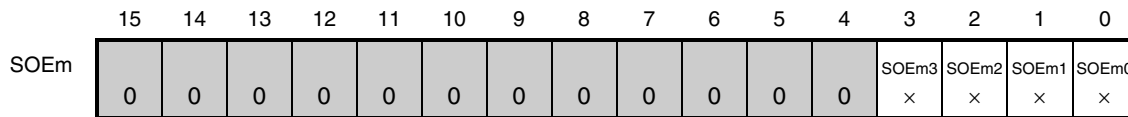
(1) Register setting

Figure 13-56. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

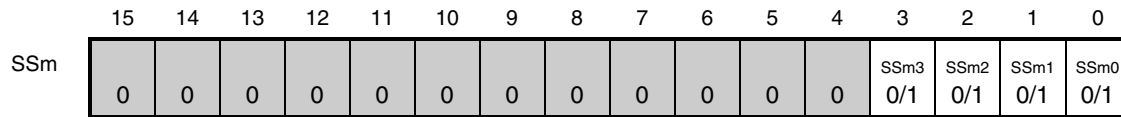
(a) Serial output register m (SOM) ...The register that not used in this mode.



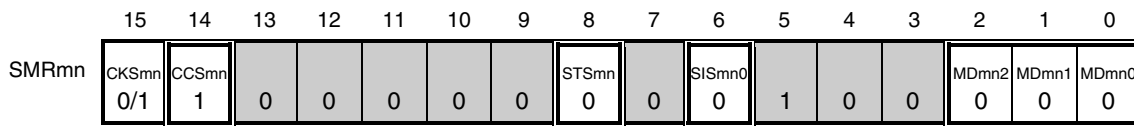
<R> (b) Serial output enable register m (SOEm) ...The register that not used in this mode.



(c) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.



(d) Serial mode register mn (SMRmn)

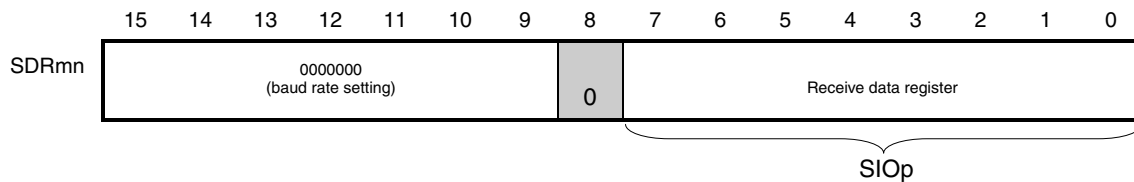


Interrupt sources of channel n  
0: Transfer end interrupt

(e) Serial communication operation setting register mn (SCRmn)



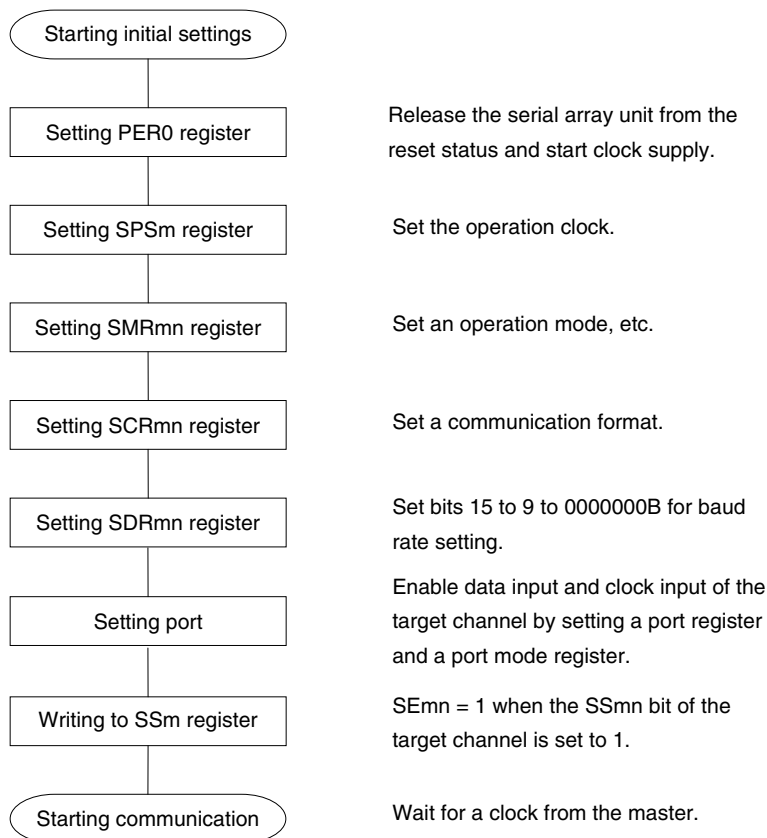
(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)  
: Setting is fixed in the CSI slave reception mode, : Setting disabled (set to the initial value)  
 x: 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

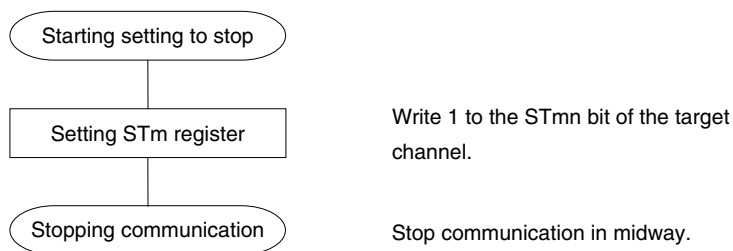
(2) Operation procedure

Figure 13-57. Initial Setting Procedure for Slave Reception



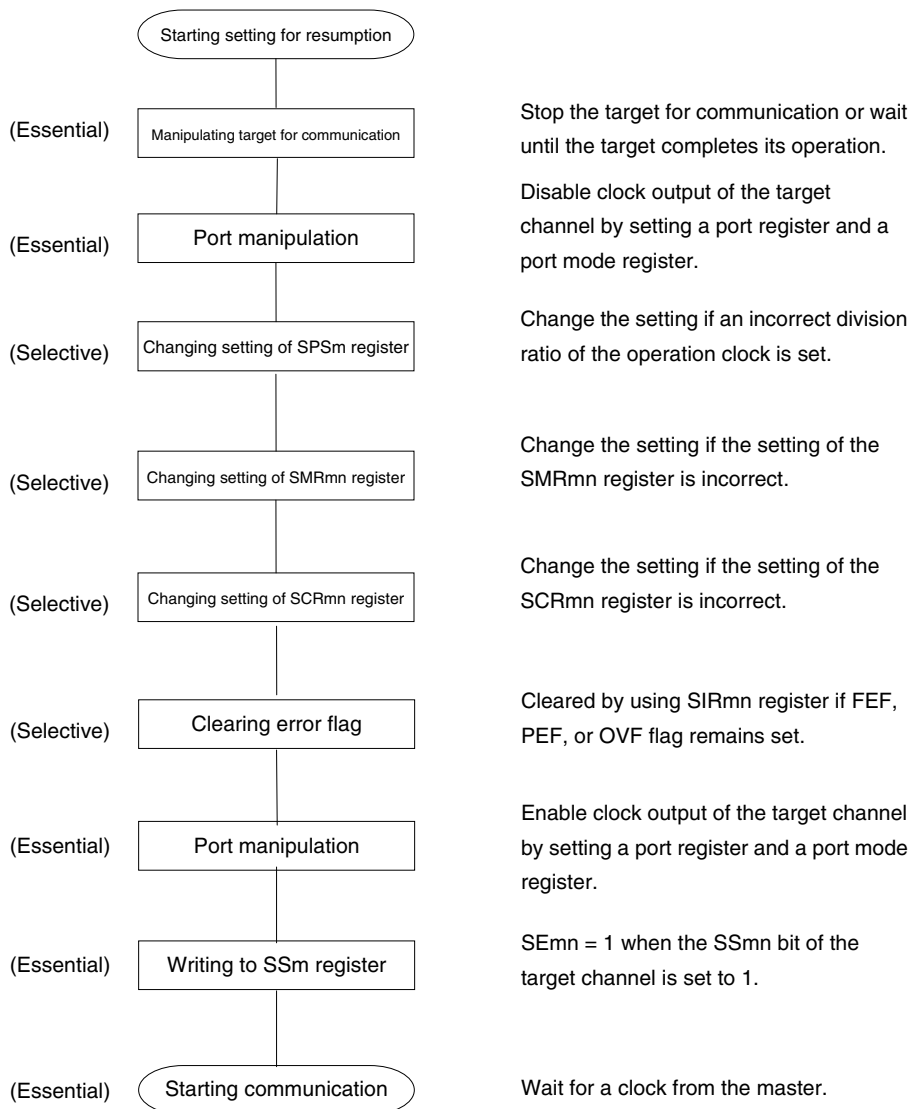
**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 13-58. Procedure for Stopping Slave Reception



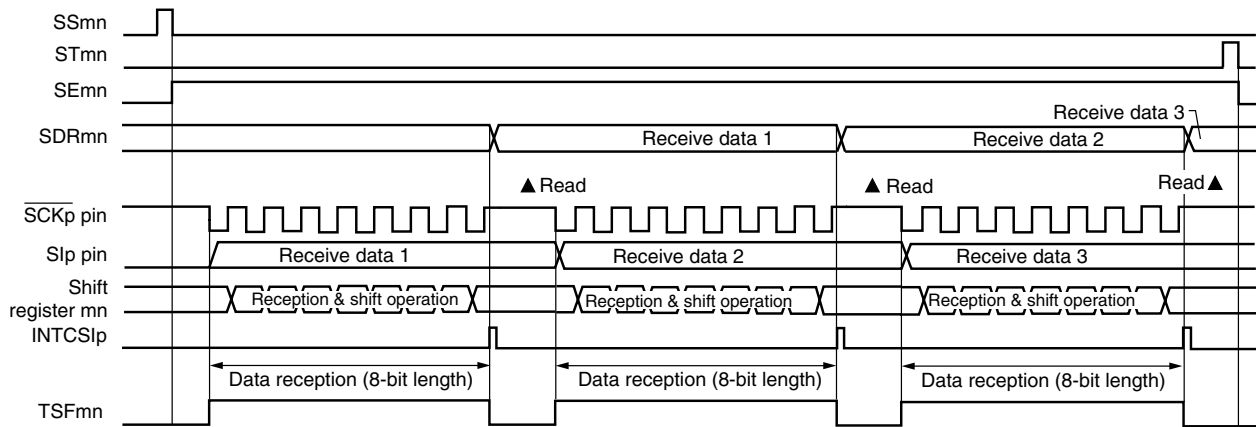
<R>

**Figure 13-59. Procedure for Resuming Slave Reception**



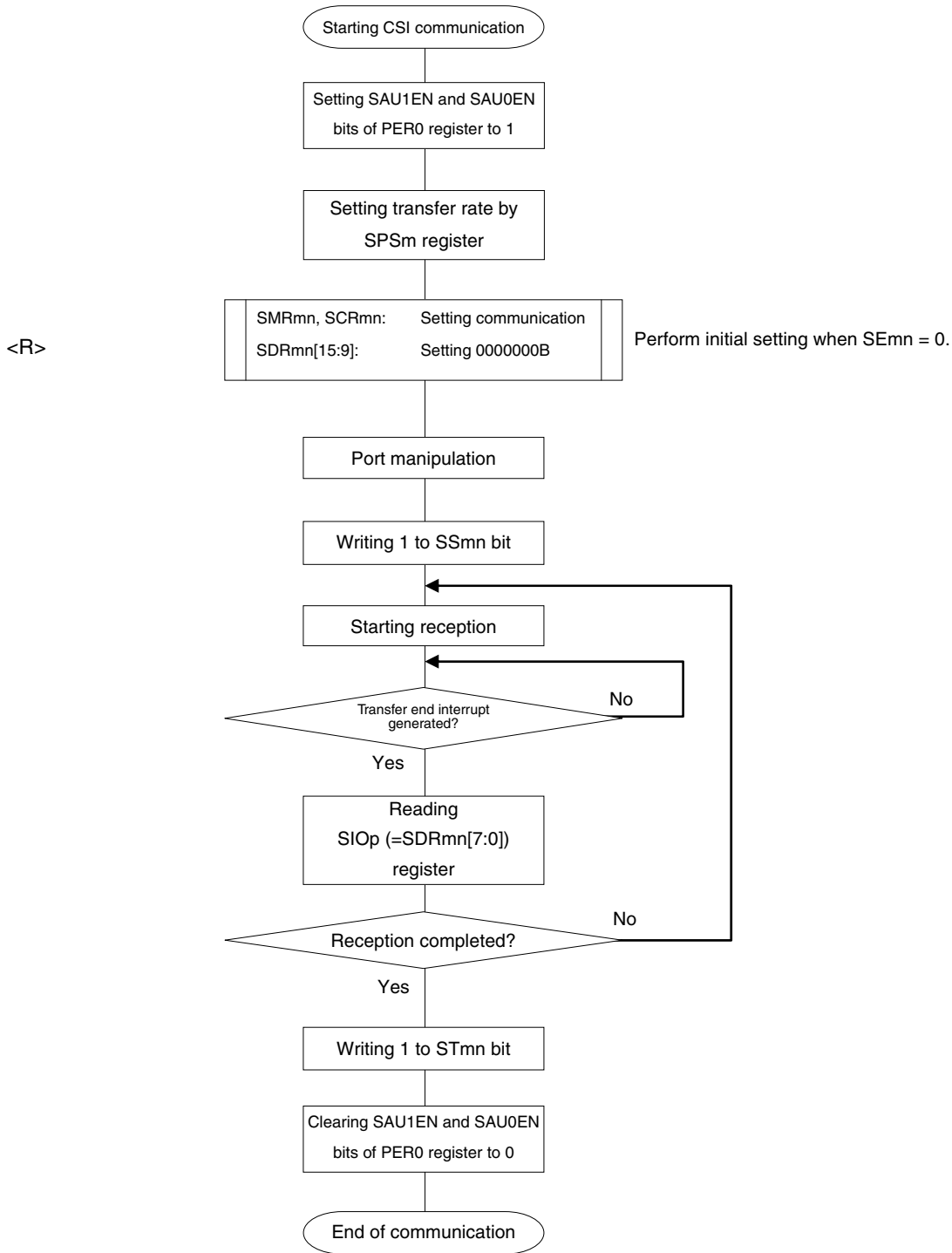
## (3) Processing flow (in single-reception mode)

Figure 13-60. Timing Chart of Slave Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

Figure 13-61. Flowchart of Slave Reception (in Single-Reception Mode)



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

### 13.5.6 Slave transmission/reception

Slave transmission/reception is that the 78K0R/KH3 transmits/receives data to/from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	$\overline{\text{SCK00}}$ , SI00, SO00	$\overline{\text{SCK01}}$ , SI01, SO01	$\overline{\text{SCK10}}$ , SI10, SO10	$\overline{\text{SCK11}}$ , SI11, SO11	$\overline{\text{SCK20}}$ , SI20, SO20	$\overline{\text{SCK21}}$ , SI21, SO21
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.					
Error detection flag	Overrun error detection flag (OVFmn) only					
Transfer data length	7 or 8 bits					
Transfer rate	Max. $f_{\text{MCK}}/6$ [Hz] <sup>Notes 1, 2</sup>					
Data phase	Selectable by DAPmn bit <ul style="list-style-type: none"> <li>• DAPmn = 0: Data output starts from the start of the operation of the serial clock.</li> <li>• DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.</li> </ul>					
Clock phase	Selectable by CKPmn bit <ul style="list-style-type: none"> <li>• CKPmn = 0: Forward</li> <li>• CKPmn = 1: Reverse</li> </ul>					
Data direction	MSB or LSB first					

**Notes 1.** Because the external serial clock input to pins  $\overline{\text{SCK00}}$ ,  $\overline{\text{SCK01}}$ ,  $\overline{\text{SCK10}}$ ,  $\overline{\text{SCK11}}$ ,  $\overline{\text{SCK20}}$ , and  $\overline{\text{SCK21}}$  is sampled internally and used, the fastest transfer rate is  $f_{\text{MCK}}/6$  [Hz].

- 2.** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)** and **CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)**).

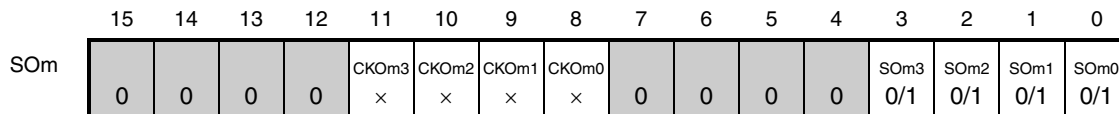
**Remarks 1.**  $f_{\text{MCK}}$ : Operation clock (MCK) frequency of target channel

- 2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

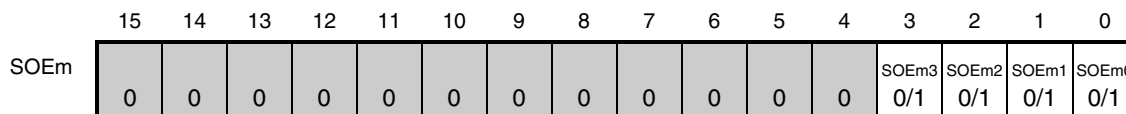
(1) Register setting

Figure 13-62. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)

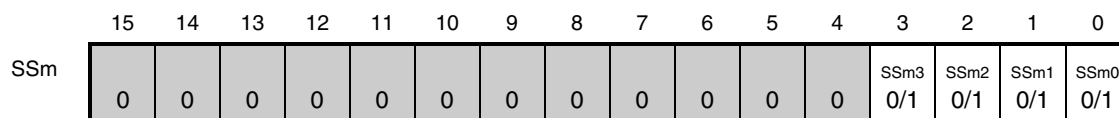
(a) Serial output register m (SOM) ... Sets only the bits of the target channel.



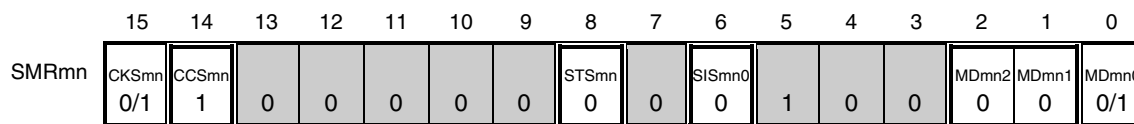
(b) Serial output enable register m (SOEm) ... Sets only the bits of the target channel to 1.



(c) Serial channel start register m (SSm) ... Sets only the bits of the target channel to 1.

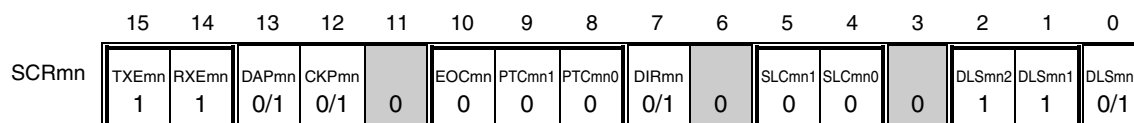


(d) Serial mode register mn (SMRmn)

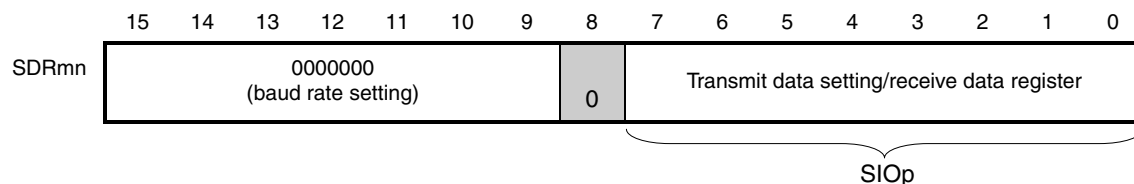


Interrupt sources of channel n  
 0: Transfer end interrupt  
 1: Buffer empty interrupt

(e) Serial communication operation setting register mn (SCRmn)



(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



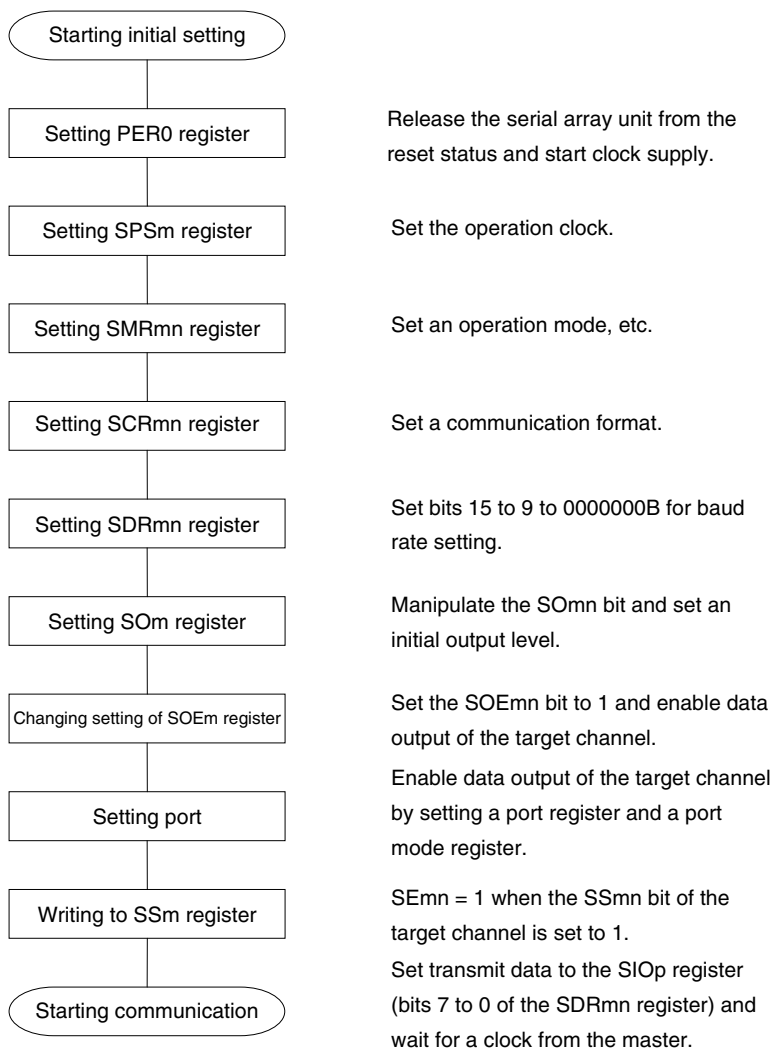
<R> **Caution** Be sure to set transmit data to the SIOp register before the clock from the master is started.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)  
 □ : Setting is fixed in the CSI slave transmission/reception mode, ■ : Setting disabled (set to the initial value)  
 ×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)  
 0/1: Set to 0 or 1 depending on the usage of the user



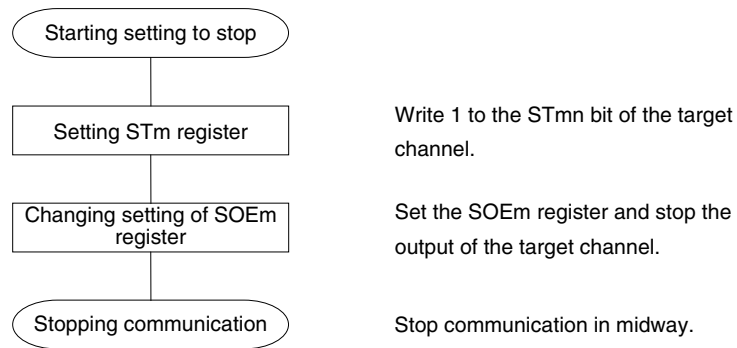
## (2) Operation procedure

Figure 13-63. Initial Setting Procedure for Slave Transmission/Reception



- Cautions**
1. After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.
  2. Be sure to set transmit data to the SIOp register before the clock from the master is started.

&lt;R&gt;

**Figure 13-64. Procedure for Stopping Slave Transmission/Reception**

**Remark** Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 13-65 Procedure for Resuming Slave Transmission/Reception**).

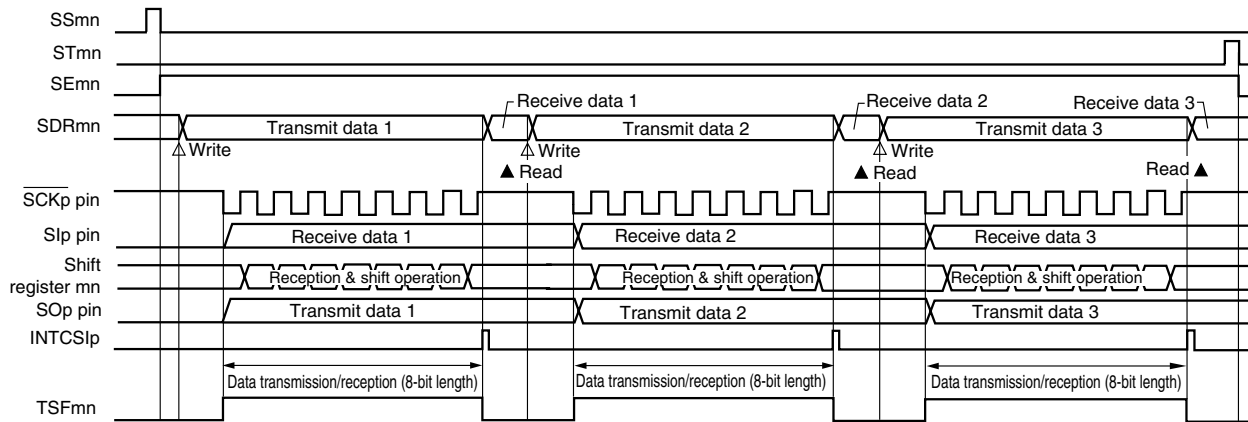
Figure 13-65. Procedure for Resuming Slave Transmission/Reception



<R> **Caution** Be sure to set transmit data to the SIOp register before the clock from the master is started.

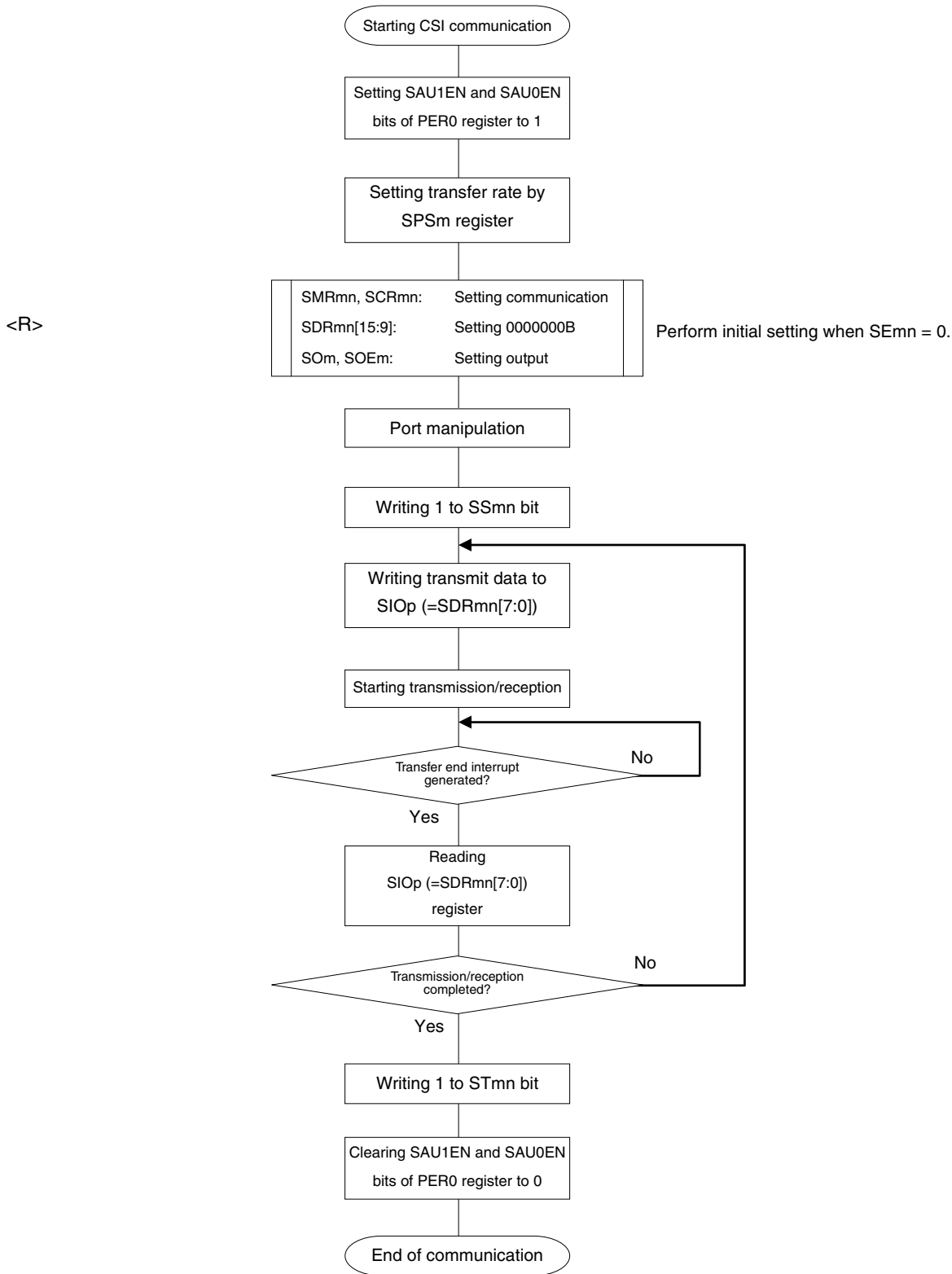
(3) Processing flow (in single-transmission/reception mode)

Figure 13-66. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

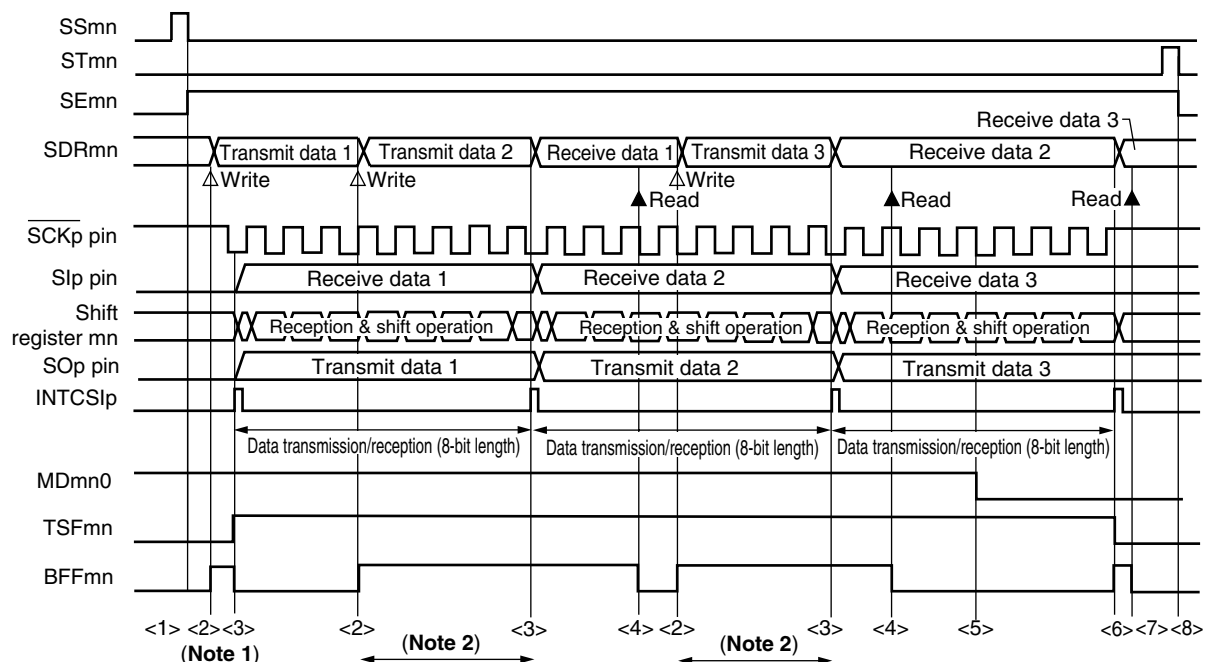
Figure 13-67. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)



- <R>
- Cautions 1.** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.
- 2.** 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 13-68. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)  
(Type 1: DAPmn = 0, CKPmn = 0)



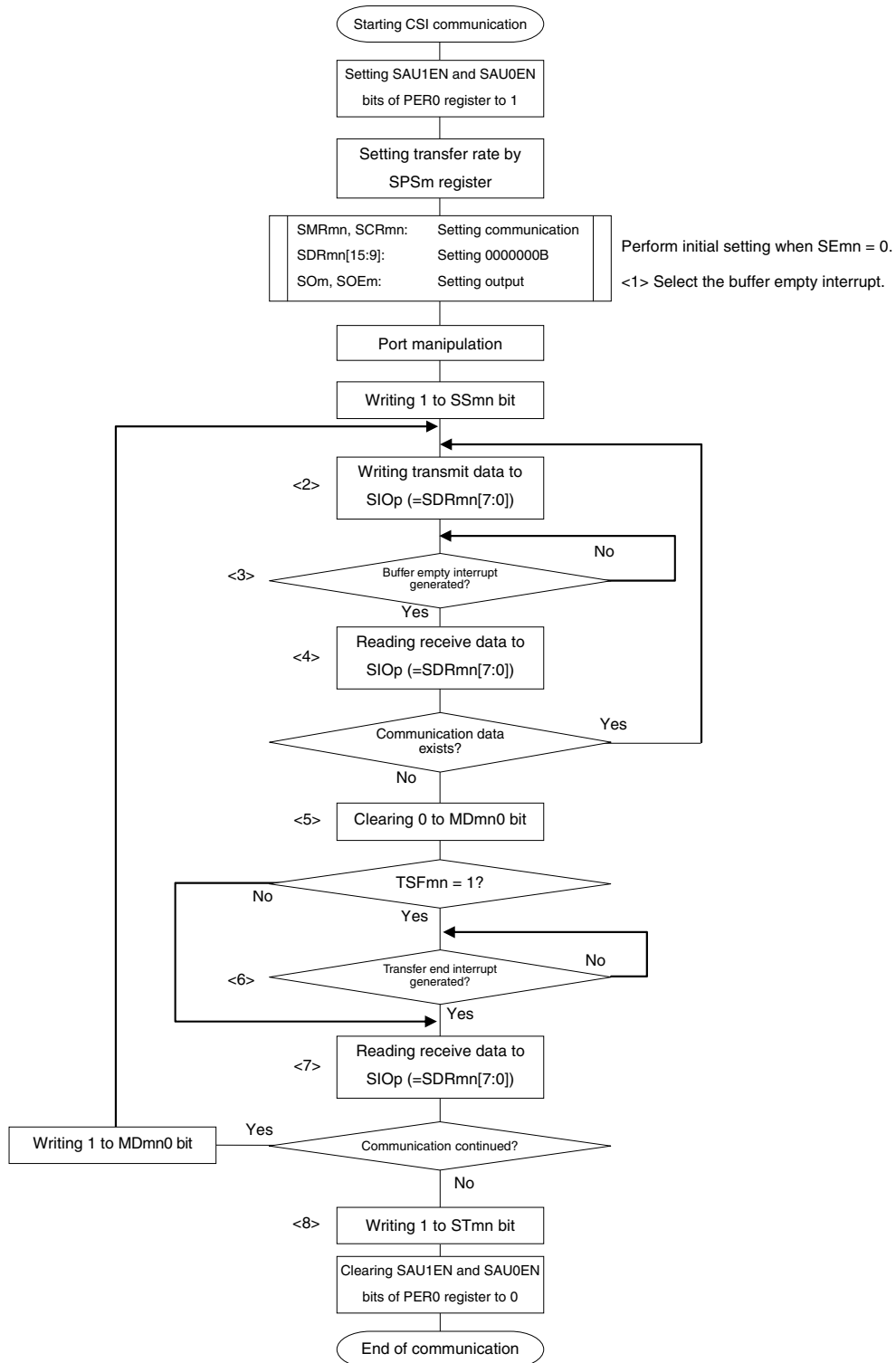
- Notes 1.** When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.
- 2.** The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

**Caution** The MDmn0 bit 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.

- Remarks 1.** <1> to <8> in the figure correspond to <1> to <8> in Figure 13-69 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).
- 2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21)

Figure 13-69. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)

<R>



**Cautions 1.** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

**2.** 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 13-68 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

**13.5.7 Calculating transfer clock frequency**

The transfer clock frequency for 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) communication can be calculated by the following expressions.

**(1) Master**

$$(\text{Transfer clock frequency}) = \{\text{Operation clock (MCK) frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2 \text{ [Hz]}$$

**(2) Slave**

$$(\text{Transfer clock frequency}) = \{\text{Frequency of serial clock (SCK) supplied by master}\}^{\text{Note}} \text{ [Hz]}$$

**Note** The permissible maximum transfer clock frequency is  $f_{\text{MCK}}/6$ .

**Remarks 1.** The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000000B to 1111111B) and therefore is 0 to 127.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).



Table 13-2. Operating Clock Selection

SMRmn Register	SPSm Register								Operation Clock (MCK) <sup>Note1</sup>	
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	f <sub>CLK</sub> = 20 MHz
0	X	X	X	X	0	0	0	0	f <sub>CLK</sub>	20 MHz
	X	X	X	X	0	0	0	1	f <sub>CLK</sub> /2	10 MHz
	X	X	X	X	0	0	1	0	f <sub>CLK</sub> /2 <sup>2</sup>	5 MHz
	X	X	X	X	0	0	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	2.5 MHz
	X	X	X	X	0	1	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	1.25 MHz
	X	X	X	X	0	1	0	1	f <sub>CLK</sub> /2 <sup>5</sup>	625 kHz
	X	X	X	X	0	1	1	0	f <sub>CLK</sub> /2 <sup>6</sup>	313 kHz
	X	X	X	X	0	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	156 kHz
	X	X	X	X	1	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	78.1 kHz
	X	X	X	X	1	0	0	1	f <sub>CLK</sub> /2 <sup>9</sup>	39.1 kHz
	X	X	X	X	1	0	1	0	f <sub>CLK</sub> /2 <sup>10</sup>	19.5 kHz
	X	X	X	X	1	0	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0, INTTM03 if m = 1 <sup>Note2</sup>	
1	0	0	0	0	X	X	X	X	f <sub>CLK</sub>	20 MHz
	0	0	0	1	X	X	X	X	f <sub>CLK</sub> /2	10 MHz
	0	0	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>2</sup>	5 MHz
	0	0	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>3</sup>	2.5 MHz
	0	1	0	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>4</sup>	1.25 MHz
	0	1	0	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>5</sup>	625 kHz
	0	1	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>6</sup>	313 kHz
	0	1	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>7</sup>	156 kHz
	1	0	0	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>8</sup>	78.1 kHz
	1	0	0	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>9</sup>	39.1 kHz
	1	0	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>10</sup>	19.5 kHz
	1	0	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>11</sup>	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0, INTTM03 if m = 1 <sup>Note2</sup>	
Other than above									Setting prohibited	

**Notes 1.** When changing the clock selected for f<sub>CLK</sub> (by changing the system clock control register (CKC) value), do so after having stopped (STm = 000FH) the operation of the serial array unit (SAU). When selecting INTTM02 and INTTM03 for the operation clock, also stop the timer array unit 0 (TAU0) (TT0 = 00FFH).

- 2.** SAU0 can be operated at a fixed division ratio of the subsystem clock, regardless of the f<sub>CLK</sub> frequency (main system clock, subsystem clock), by operating the interval timer for which f<sub>SUB</sub>/4 has been selected as the count clock (setting TIS02 (if m = 0) or TIS03 (if m = 1) of the TIS0 register to 1) and selecting INTTM02 and INTTM03 by using the SPSm register in channels 2 and 3 of TAU. When changing f<sub>CLK</sub>, however, SAU and TAU0 must be stopped as described in Note 1 above.

**Remarks 1.** X: Don't care

- 2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

### 13.5.8 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) communication

The procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) communication is described in Figure 13-70.

**Figure 13-70. Processing Procedure in Case of Overrun Error**

Software Manipulation	Hardware Status	Remark
Reads serial data SDRmn register. —————>	The BFF = 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register —————>	Error flag is cleared.	Only error generated at the point of reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

### 13.6 Operation of UART (UART0, UART1, UART2, UART3) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. It transmits or receives data in asynchronization with the party of communication (by using an internal baud rate). Full-duplex UART communication can be realized by using two channels, one dedicated to transmission (even channel) and the other to reception (odd channel).

[Data transmission/reception]

- Data length of 5, 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

The LIN-bus is supported in UART3 (2, 3 channels of unit 1)

[LIN-bus functions]

- Wakeup signal detection
  - Sync break field (SBF) detection
  - Sync field measurement, baud rate calculation
- } External interrupt (INTP0) or timer array unit (TAU) is used.

UART0 uses channels 0 and 1 of SAU0.

UART1 uses channels 2 and 3 of SAU0.

UART2 uses channels 0 and 1 of SAU1.

UART3 uses channels 2 and 3 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I <sup>2</sup> C
0	0	CSI00	UART0	–
	1	CSI01		–
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

**Caution** When using serial array units 0 and 1 as UARTs, the channels of both the transmitting side (even-number channel) and the receiving side (odd-number channel) can be used only as UARTs.

UART performs the following four types of communication operations.

- UART transmission (See 13.6.1.)
- UART reception (See 13.6.2.)
- LIN transmission (UART3 only) (See 13.6.3.)
- LIN reception (UART 3 only) (See 13.6.4.)

### 13.6.1 UART transmission

UART transmission is an operation to transmit data from the 78K0R/KH3 to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

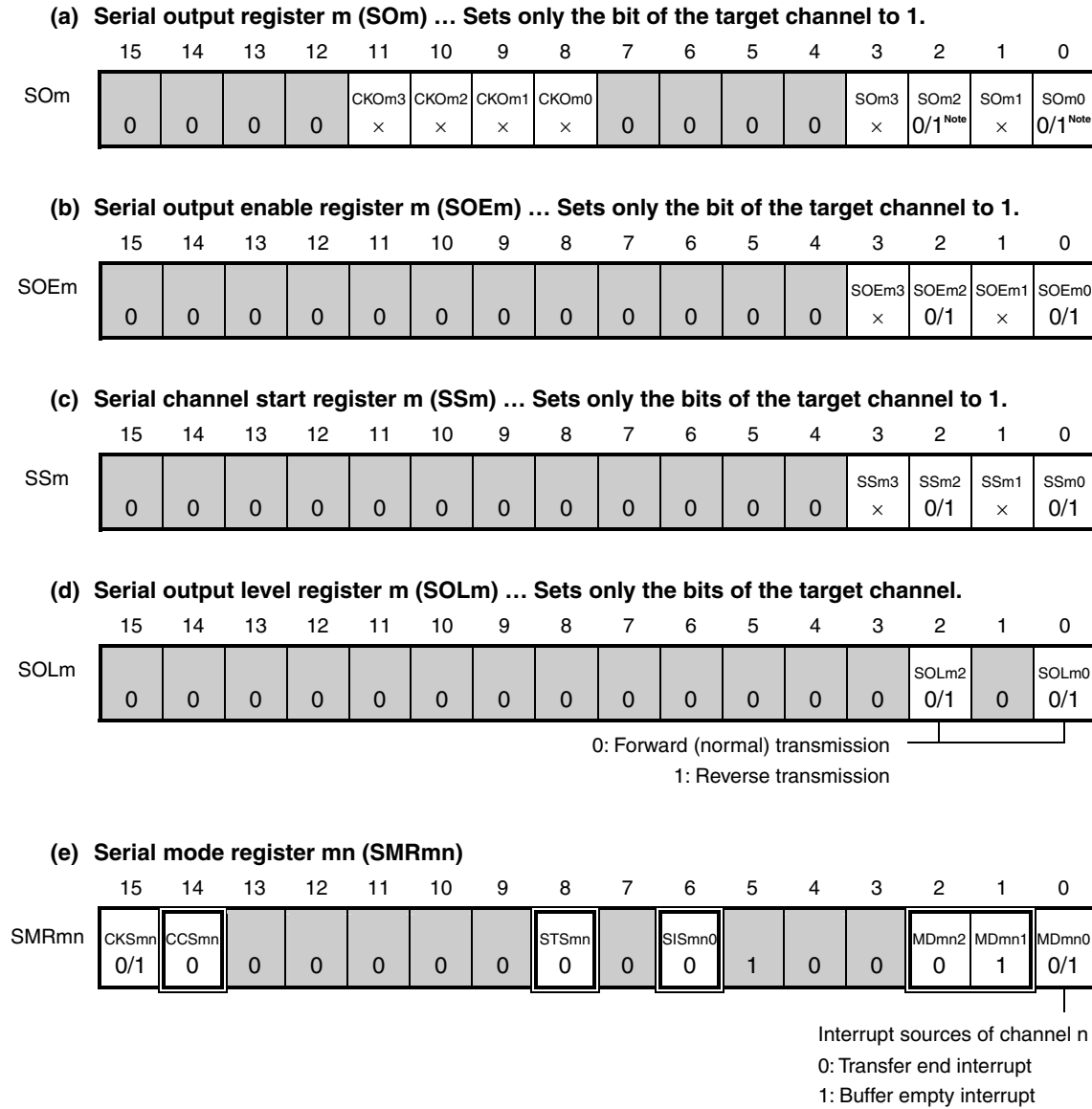
UART	UART0	UART1	UART2	UART3
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	TxD0	TxD1	TxD2	TxD3
Interrupt	INTST0	INTST1	INTST2	INTST3
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	None			
Transfer data length	5, 7, or 8 bits			
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR <sub>mn</sub> [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] <sup>Note</sup>			
Data phase	Forward output (default: high level) Reverse output (default: low level)			
Parity bit	The following selectable <ul style="list-style-type: none"> <li>• No parity bit</li> <li>• Appending 0 parity</li> <li>• Appending even parity</li> <li>• Appending odd parity</li> </ul>			
Stop bit	The following selectable <ul style="list-style-type: none"> <li>• Appending 1 bit</li> <li>• Appending 2 bits</li> </ul>			
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 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)** and **CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)**).

**Remarks** 1.  $f_{MCK}$ : Operation clock (MCK) frequency of target channel  
 $f_{CLK}$ : System clock frequency  
2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(1) Register setting

Figure 13-71. Example of Contents of Registers for UART Transmission of UART (UART0, UART1, UART2, UART3) (1/2)

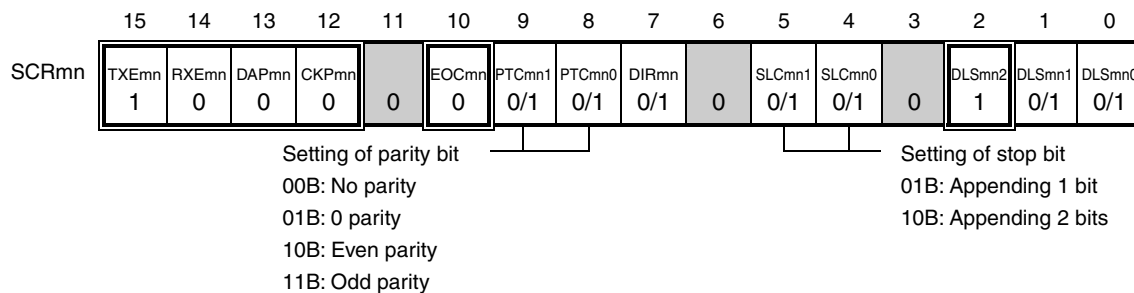


**Note** Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

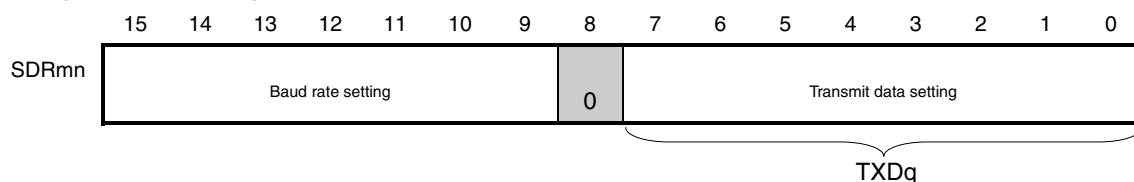
**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)  
: Setting is fixed in the UART transmission mode, : Setting disabled (set to the initial value)  
 ×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)  
 0/1: Set to 0 or 1 depending on the usage of the user

Figure 13-71. Example of Contents of Registers for UART Transmission of UART (UART0, UART1, UART2, UART3) (2/2)

(f) Serial communication operation setting register mn (SCRmn)



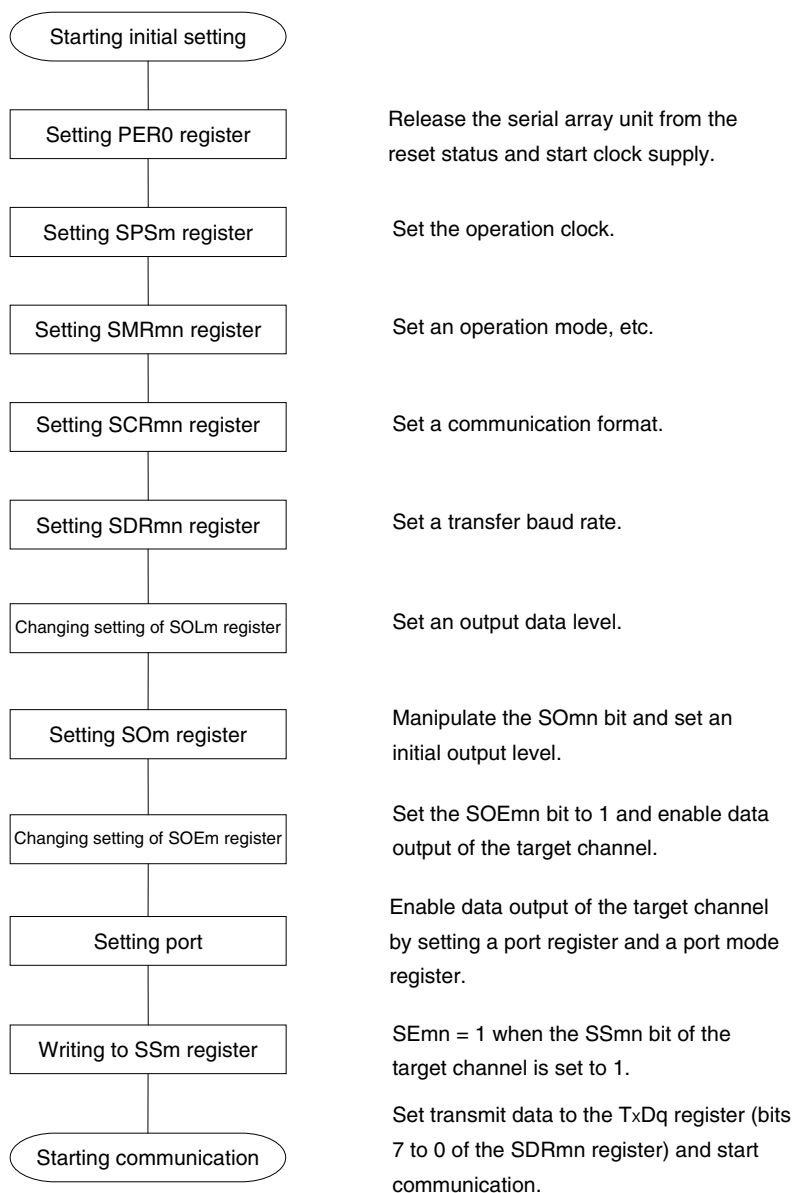
(g) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



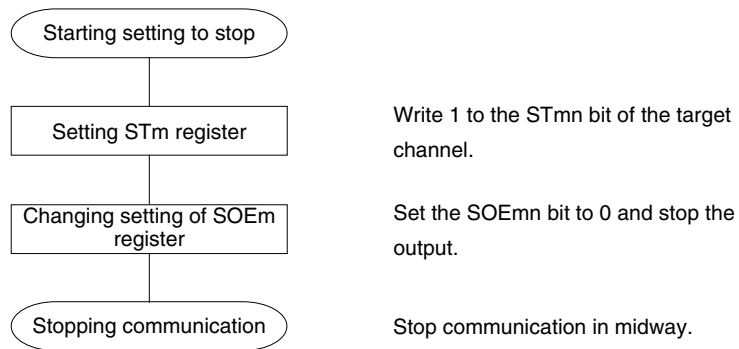
**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)  
: Setting is fixed in the UART transmission mode, : Setting disabled (set to the initial value)  
 0/1: Set to 0 or 1 depending on the usage of the user

## (2) Operation procedure

Figure 13-72. Initial Setting Procedure for UART Transmission



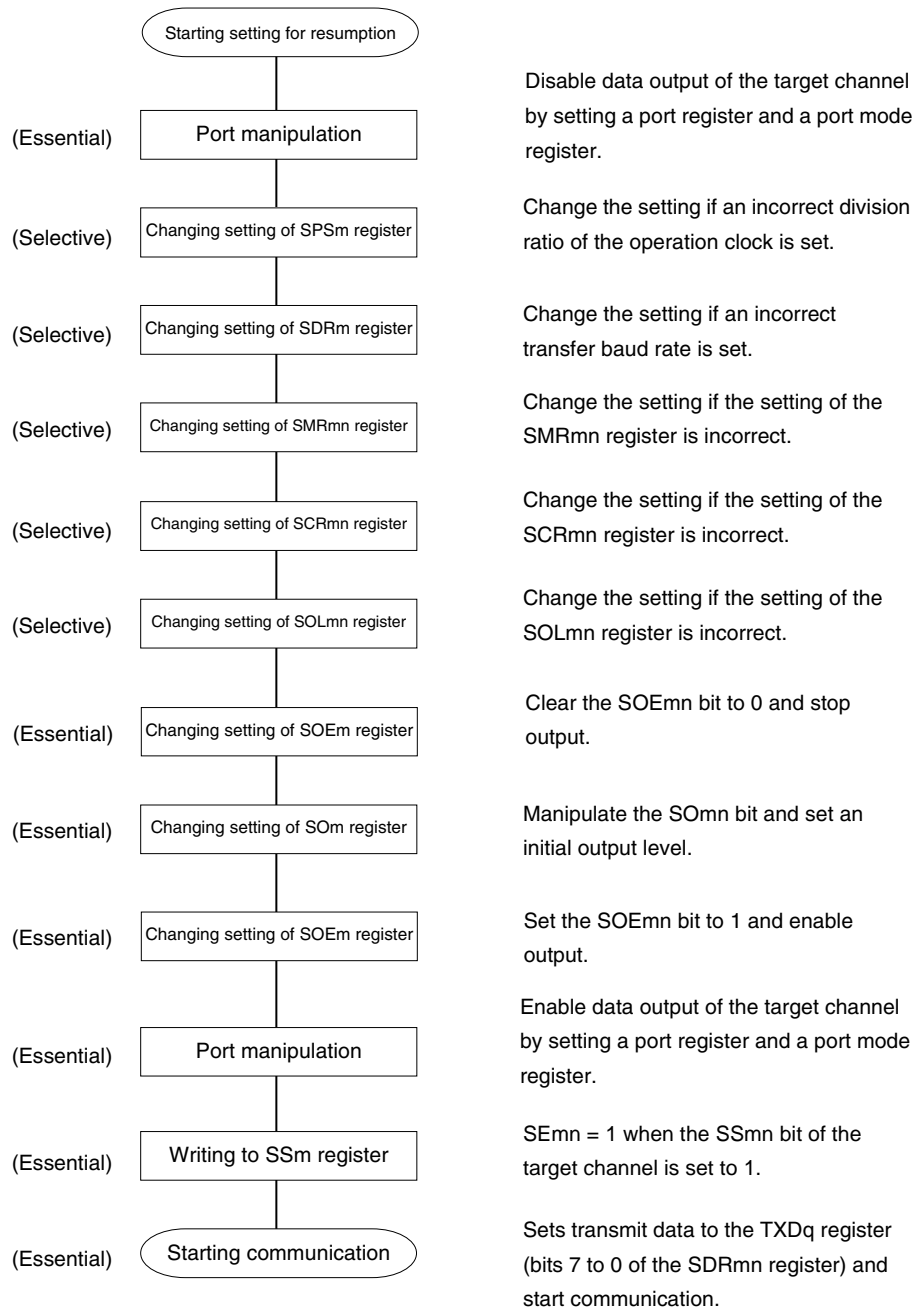
**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

**Figure 13-73. Procedure for Stopping UART Transmission**

**Remark** Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 13-74 Procedure for Resuming UART Transmission**).

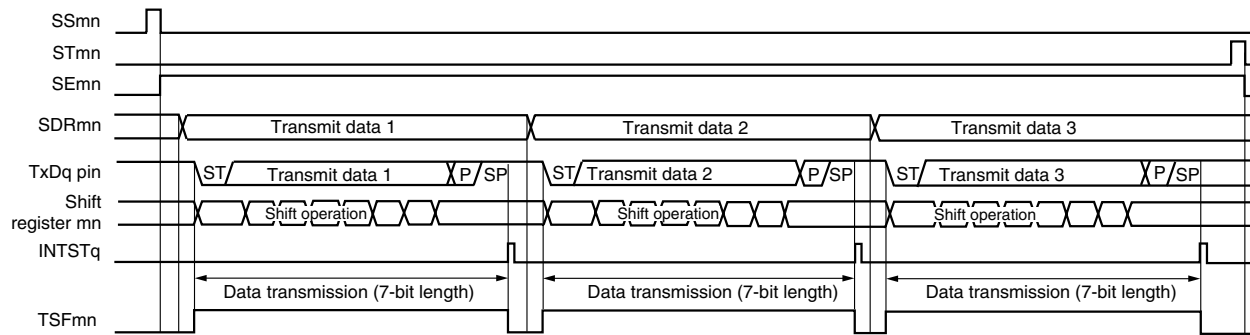


**Figure 13-74. Procedure for Resuming UART Transmission**



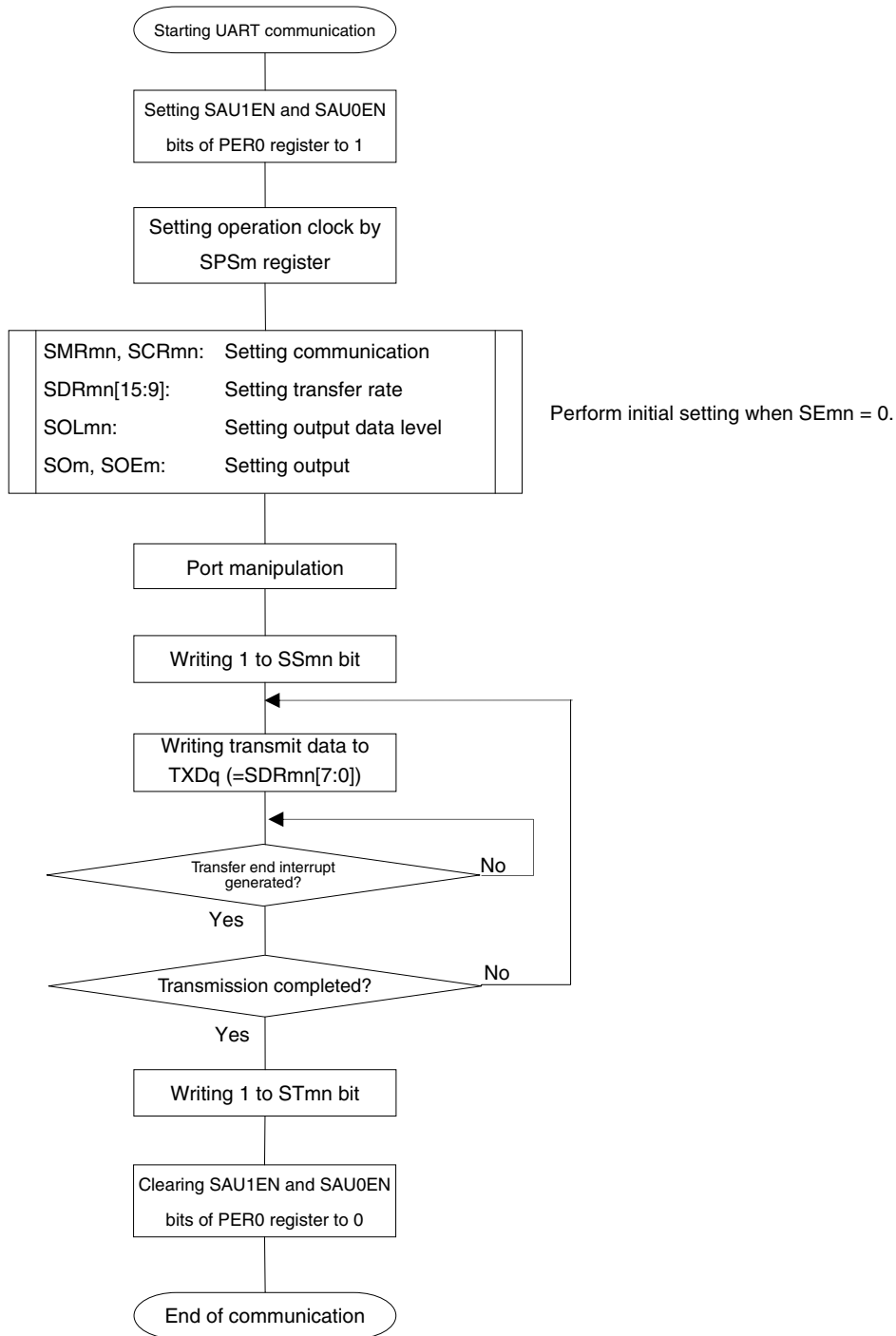
(3) Processing flow (in single-transmission mode)

Figure 13-75. Timing Chart of UART Transmission (in Single-Transmission Mode)



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), q: UART number (q = 0 to 3)

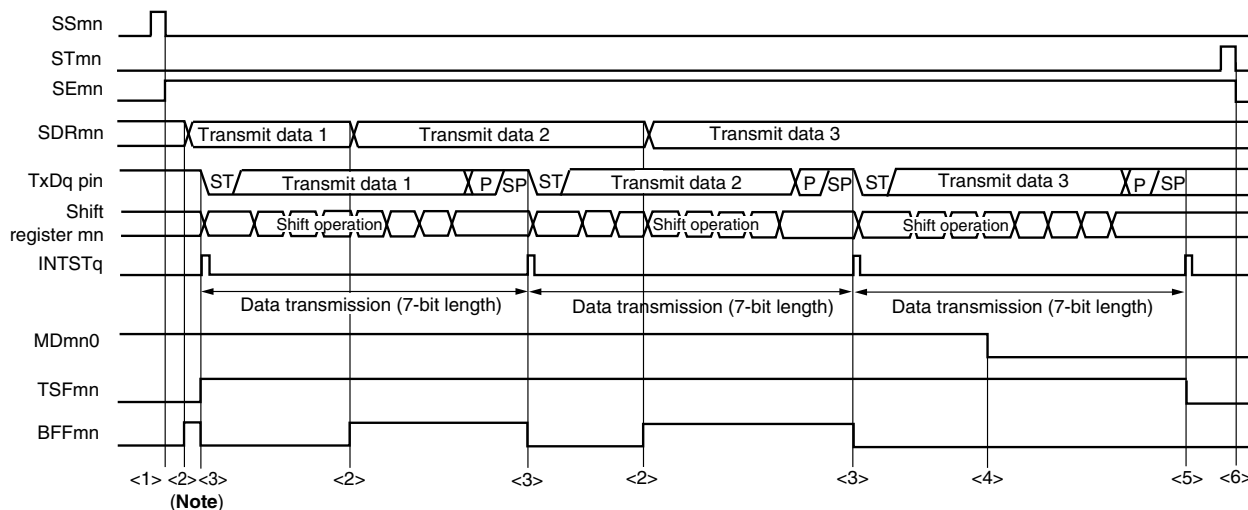
Figure 13-76. Flowchart of UART Transmission (in Single-Transmission Mode)



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 13-77. Timing Chart of UART Transmission (in Continuous Transmission Mode)

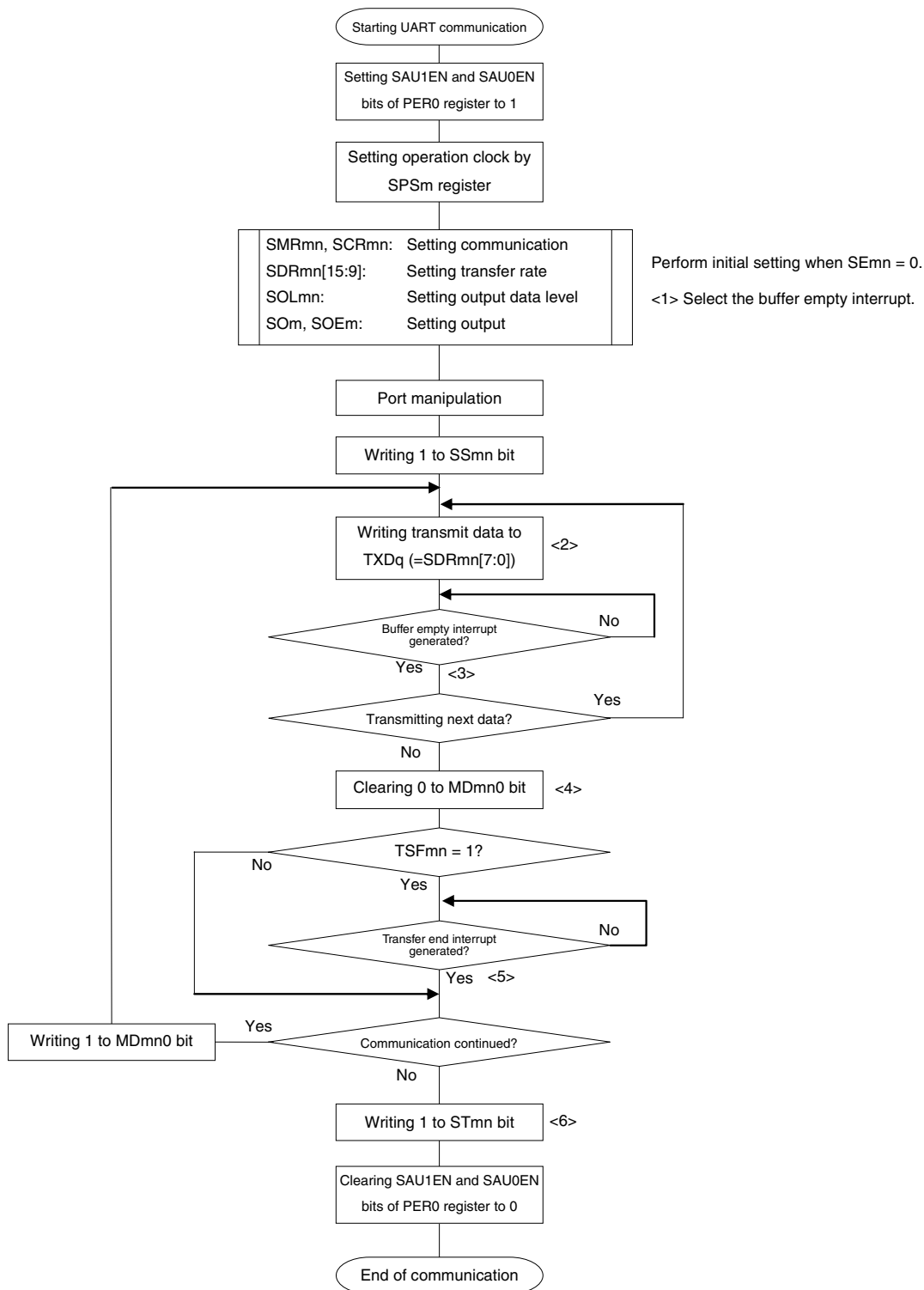


**Note** When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

**Caution** The MDmn0 bit can be rewritten even during operation.  
 However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), q: UART number (q = 0 to 3)

Figure 13-78. Flowchart of UART Transmission (in Continuous Transmission Mode)



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

**Remark** <1> to <6> in the figure correspond to <1> to <6> in Figure 13-77 Timing Chart of UART Transmission (in Continuous Transmission Mode).

### 13.6.2 UART reception

UART reception is an operation wherein the 78K0R/KH3 asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMR register of both the odd- and even-numbered channels must be set.

UART	UART0	UART1	UART2	UART3
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1	Channel 3 of SAU1
Pins used	RxD0	RxD1	RxD2	RxD3
Interrupt	INTSR0	INTSR1	INTSR2	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	INTSRE0	INTSRE1	INTSRE2	INTSRE3
Error detection flag	<ul style="list-style-type: none"> <li>• Framing error detection flag (FEF<sub>mn</sub>)</li> <li>• Parity error detection flag (PEF<sub>mn</sub>)</li> <li>• Overrun error detection flag (OVF<sub>mn</sub>)</li> </ul>			
Transfer data length	5, 7 or 8 bits			
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR <sub>mn</sub> [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] <sup>Note</sup>			
Data phase	Forward output (default: high level) Reverse output (default: low level)			
Parity bit	The following selectable <ul style="list-style-type: none"> <li>• No parity bit (no parity check)</li> <li>• Appending 0 parity (no parity check)</li> <li>• Appending even parity</li> <li>• Appending odd parity</li> </ul>			
Stop bit	Appending 1 bit			
Data direction	MSB or LSB first			

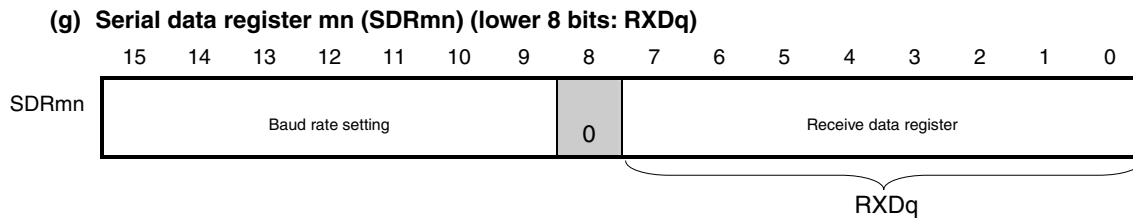
**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)** and **CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)**).

**Remarks**

1.  $f_{MCK}$ : Operation clock (MCK) frequency of target channel  
 $f_{CLK}$ : System clock frequency
2. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3)



Figure 13-79. Example of Contents of Registers for UART Reception of UART (UART0, UART1, UART2, UART3) (2/2)



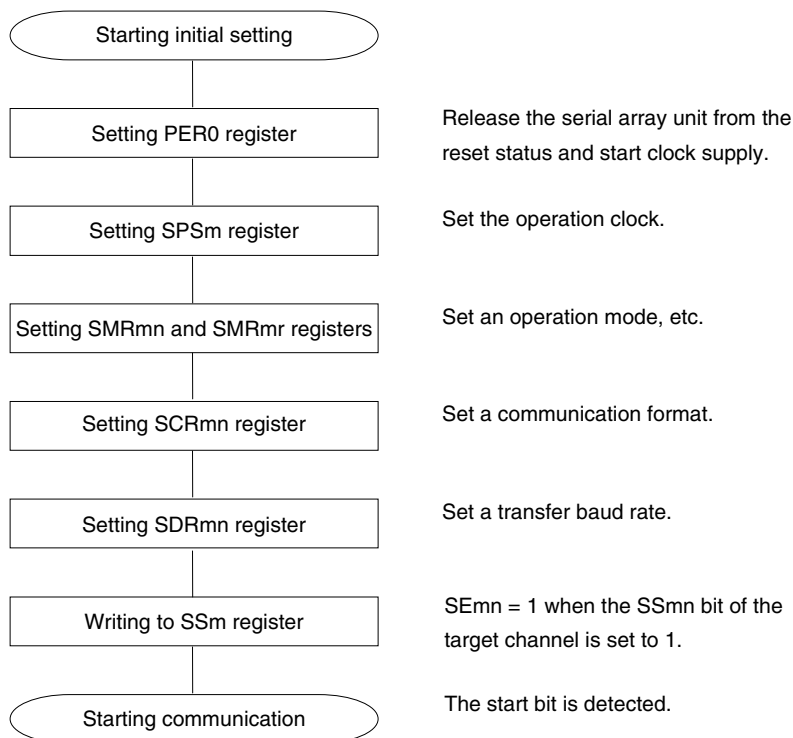
**Caution** For the UART reception, be sure to set SMRmr of channel r that is to be paired with channel n.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), r: Channel number (r = n - 1),  
 q: UART number (q = 0 to 3)  
 ■ : Setting disabled (set to the initial value)



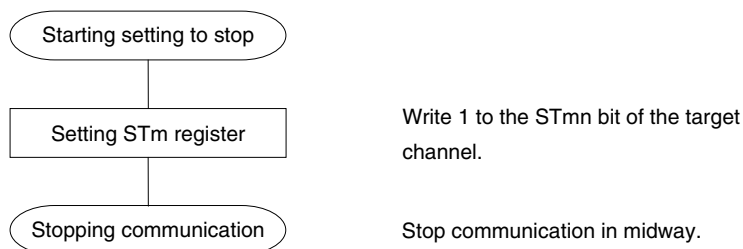
(2) Operation procedure

Figure 13-80. Initial Setting Procedure for UART Reception

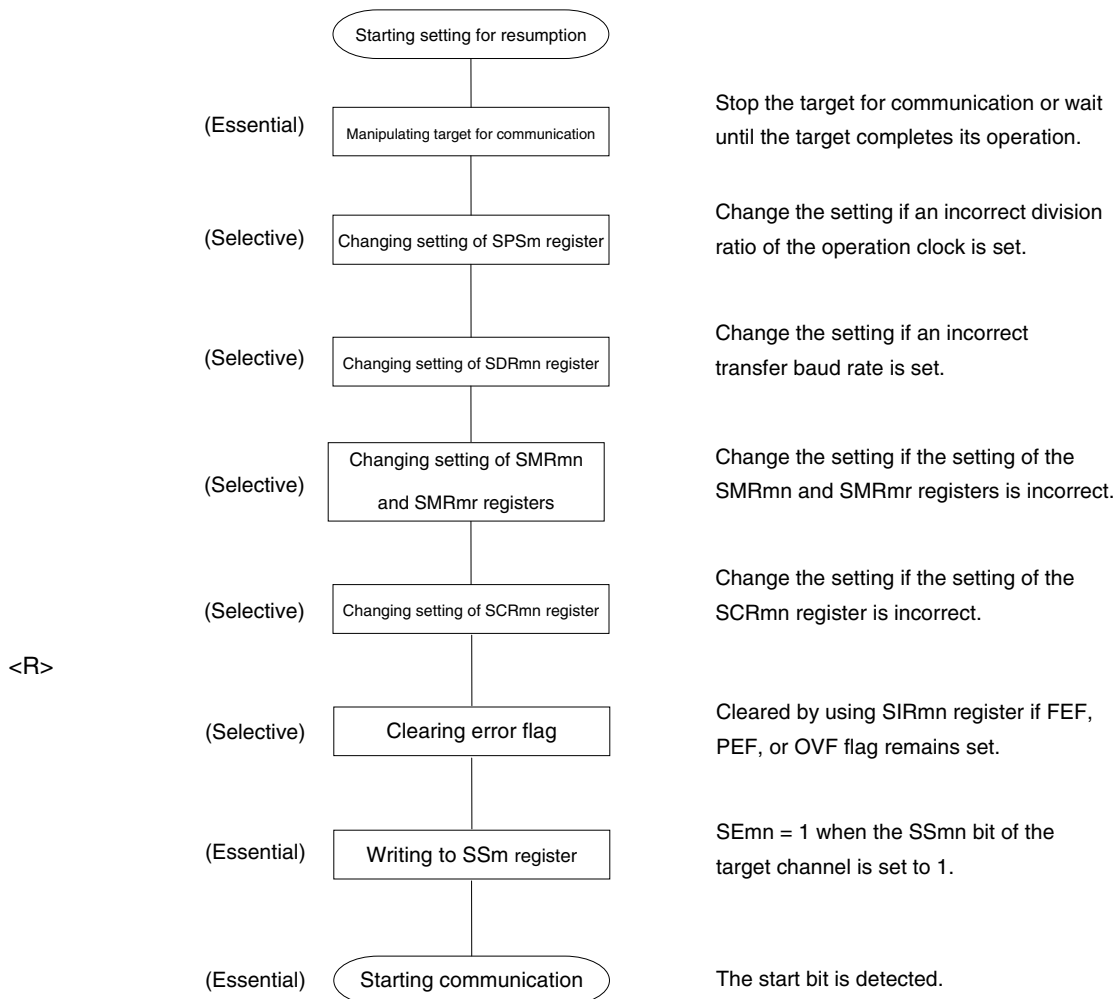


**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 13-81. Procedure for Stopping UART Reception



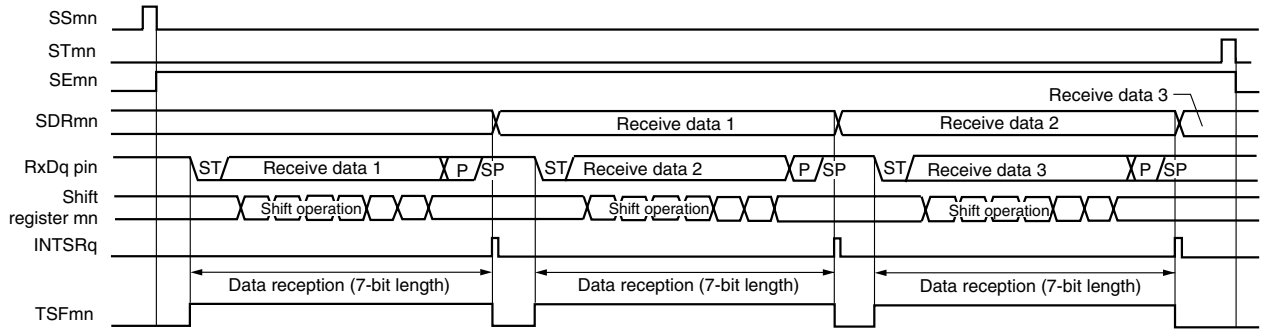
**Figure 13-82. Procedure for Resuming UART Reception**



<R>

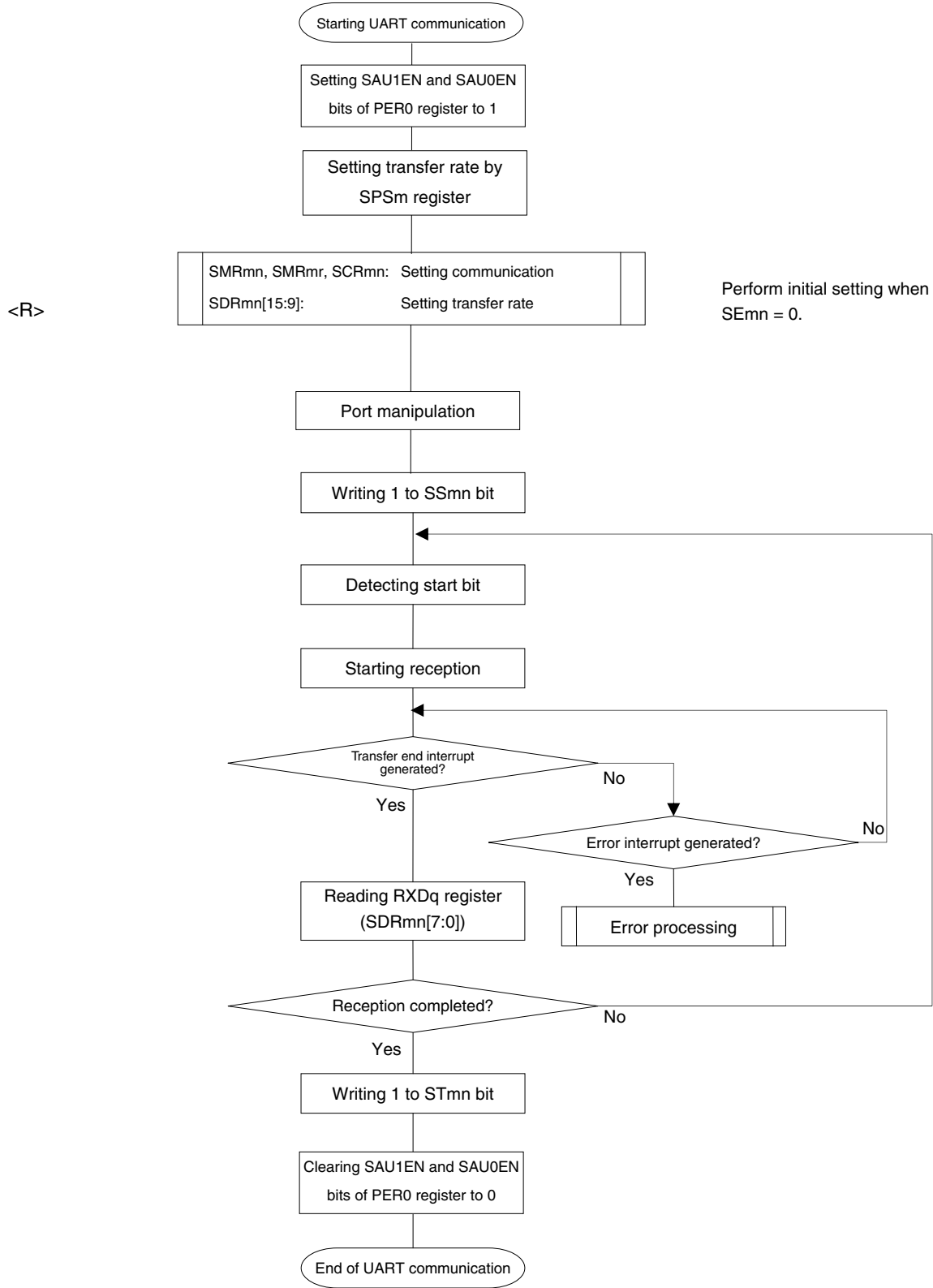
(3) Processing flow

Figure 13-83. Timing Chart of UART Reception



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), q: UART number (q = 0 to 3)

Figure 13-84. Flowchart of UART Reception



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

### 13.6.3 LIN transmission

Of UART transmission, UART3 supports LIN communication.

For LIN transmission, channel 2 of unit 1 (SAU1) is used.

UART	UART0	UART1	UART2	UART3
Support of LIN communication	Not supported	Not supported	Not supported	Supported
Target channel	–	–	–	Channel 2 of SAU1
Pins used	–	–	–	TxD3
Interrupt	–	–	–	INTST3
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	None			
Transfer data length	8 bits			
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR12 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] <sup>Note</sup>			
Data phase	Forward output (default: high level) Reverse output (default: low level)			
Parity bit	The following selectable <ul style="list-style-type: none"> <li>• No parity bit</li> <li>• Appending 0 parity</li> <li>• Appending even parity</li> <li>• Appending odd parity</li> </ul>			
Stop bit	The following selectable <ul style="list-style-type: none"> <li>• Appending 1 bit</li> <li>• Appending 2 bits</li> </ul>			
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 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)** and **CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)**).

**Remark**  $f_{MCK}$ : Operation clock (MCK) frequency of target channel  
 $f_{CLK}$ : System clock frequency

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network.

Communication of LIN is single-master communication and up to 15 slaves can be connected to one master.

The slaves are used to control switches, actuators, and sensors, which are connected to the master via LIN.

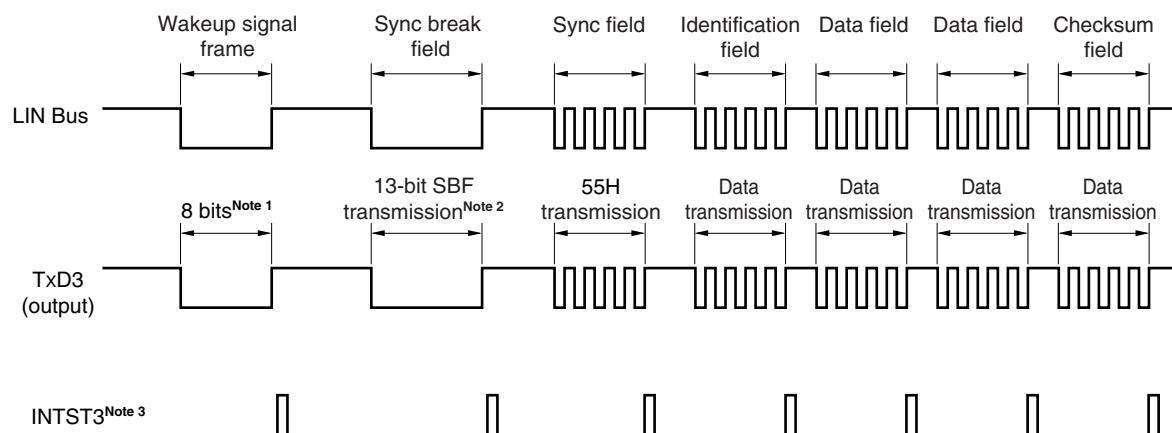
Usually, the master is connected to a network such as CAN (Controller Area Network).

A LIN bus is a single-wire bus to which nodes are connected via transceiver conforming to ISO9141.

According to the protocol of LIN, the master transmits a frame by attaching baud rate information to it. A slave receives this frame and corrects a baud rate error from the master. If the baud rate error of a slave is within  $\pm 15\%$ , communication can be established.

Figure 13-85 outlines a transmission operation of LIN.

Figure 13-85. Transmission Operation of LIN

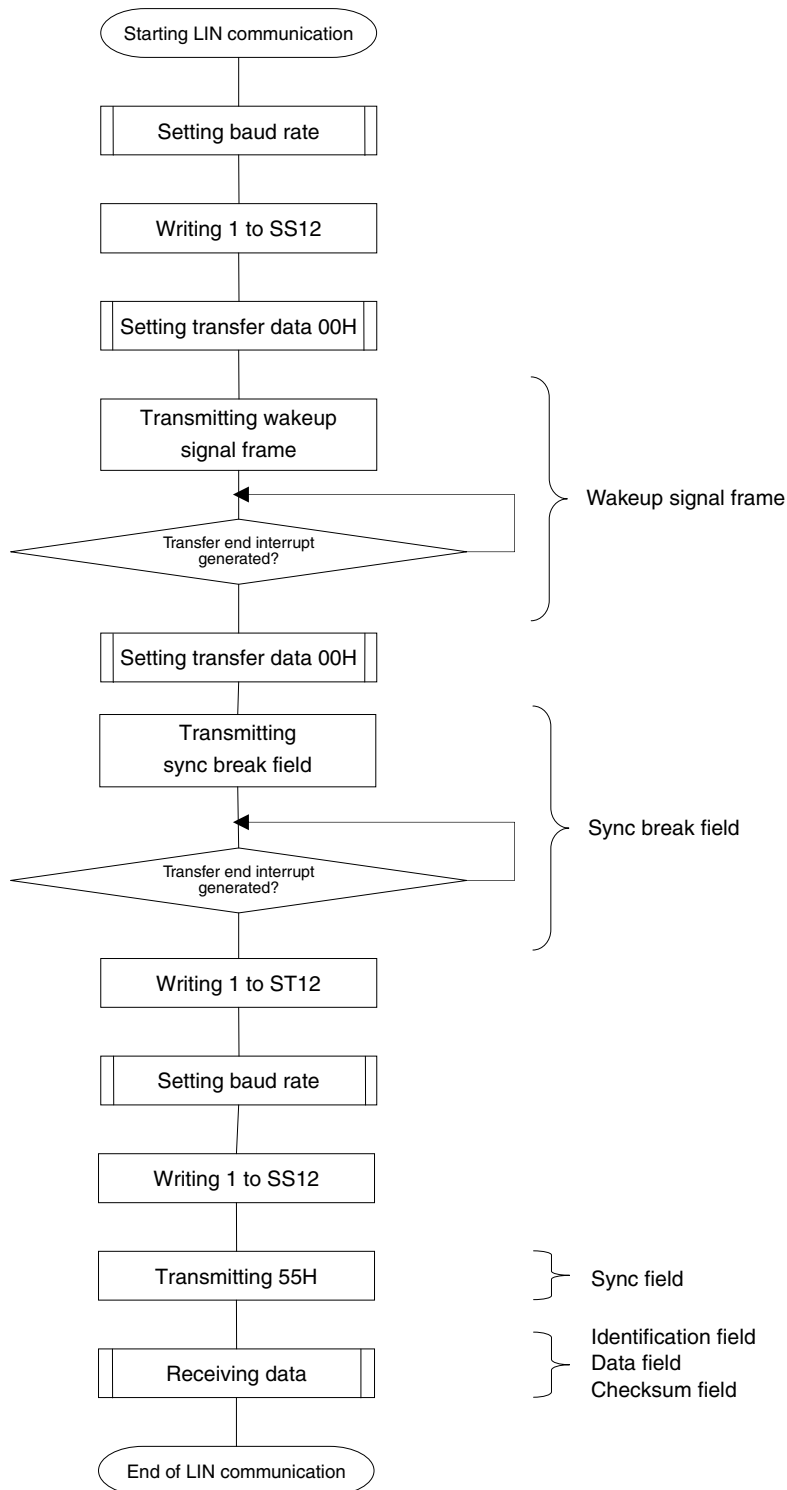


- Notes**
1. The baud rate is set so as to satisfy the standard of the wakeup signal and data of 00H is transmitted.
  2. A sync break field is defined to have a width of 13 bits and output a low level. Where the baud rate for main transfer is N [bps], therefore, the baud rate of the sync break field is calculated as follows.  

$$\text{(Baud rate of sync break field)} = 9/13 \times N$$
 By transmitting data of 00H at this baud rate, a sync break field is generated.
  3. INTST3 is output upon completion of transmission. INTST3 is also output when SBF transmission is executed.

**Remark** The interval between fields is controlled by software.

Figure 13-86. Flowchart for LIN Transmission



### 13.6.4 LIN reception

Of UART reception, UART3 supports LIN communication.

For LIN reception, channel 3 of unit 1 (SAU1) is used.

UART	UART0	UART1	UART2	UART3
Support of LIN communication	Not supported	Not supported	Not supported	Supported
Target channel	–	–	–	Channel 0 of SAU1
Pins used	–	–	–	RxD3
Interrupt	–	–	–	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	–	–	–	INTSRE3
Error detection flag	<ul style="list-style-type: none"> <li>• Framing error detection flag (FEF13)</li> <li>• Parity error detection flag (PEF13)</li> <li>• Overrun error detection flag (OVF13)</li> </ul>			
Transfer data length	8 bits			
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDR13 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] <sup>Note</sup>			
Data phase	Forward output (default: high level) Reverse output (default: low level)			
Parity bit	The following selectable <ul style="list-style-type: none"> <li>• No parity bit (no parity check)</li> <li>• Appending 0 parity (no parity check)</li> <li>• Appending even parity</li> <li>• Appending odd parity</li> </ul>			
Stop bit	The following selectable <ul style="list-style-type: none"> <li>• Appending 1 bit</li> <li>• Appending 2 bits</li> </ul>			
Data direction	MSB or LSB first			

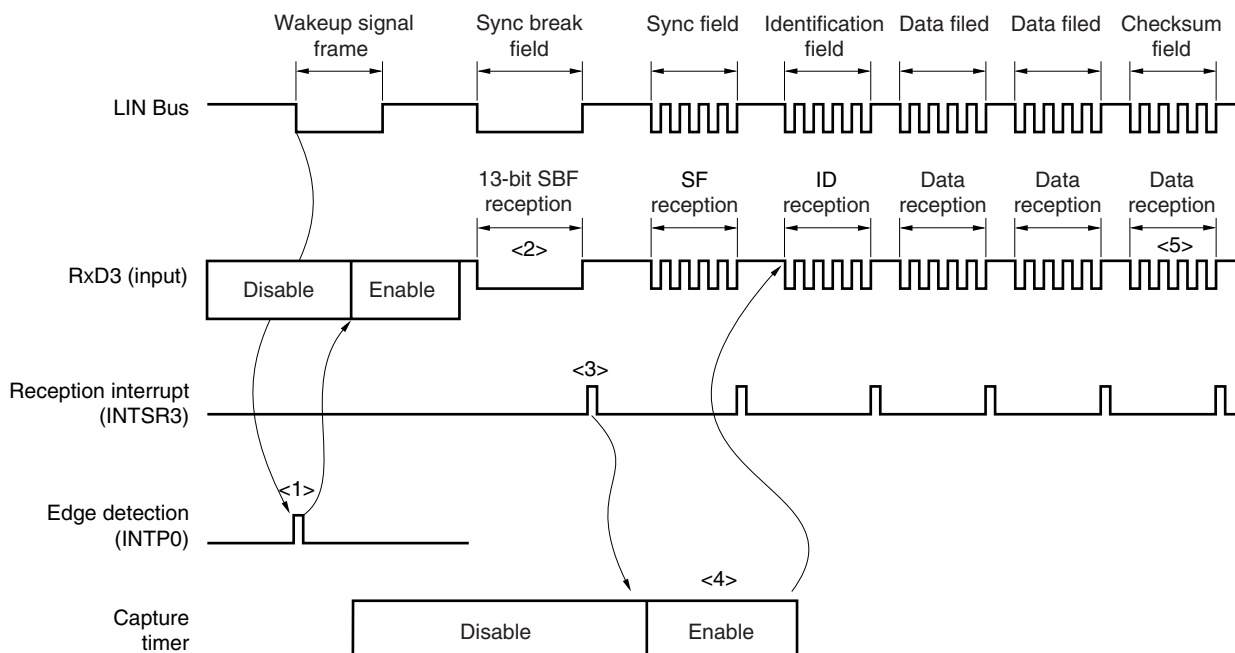
**Note** Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)** and **CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)**).

**Remark**  $f_{MCK}$ : Operation clock (MCK) frequency of target channel  
 $f_{CLK}$ : System clock frequency

Figure 13-87 outlines a reception operation of LIN.



Figure 13-87. Reception Operation of LIN



Here is the flow of signal processing.

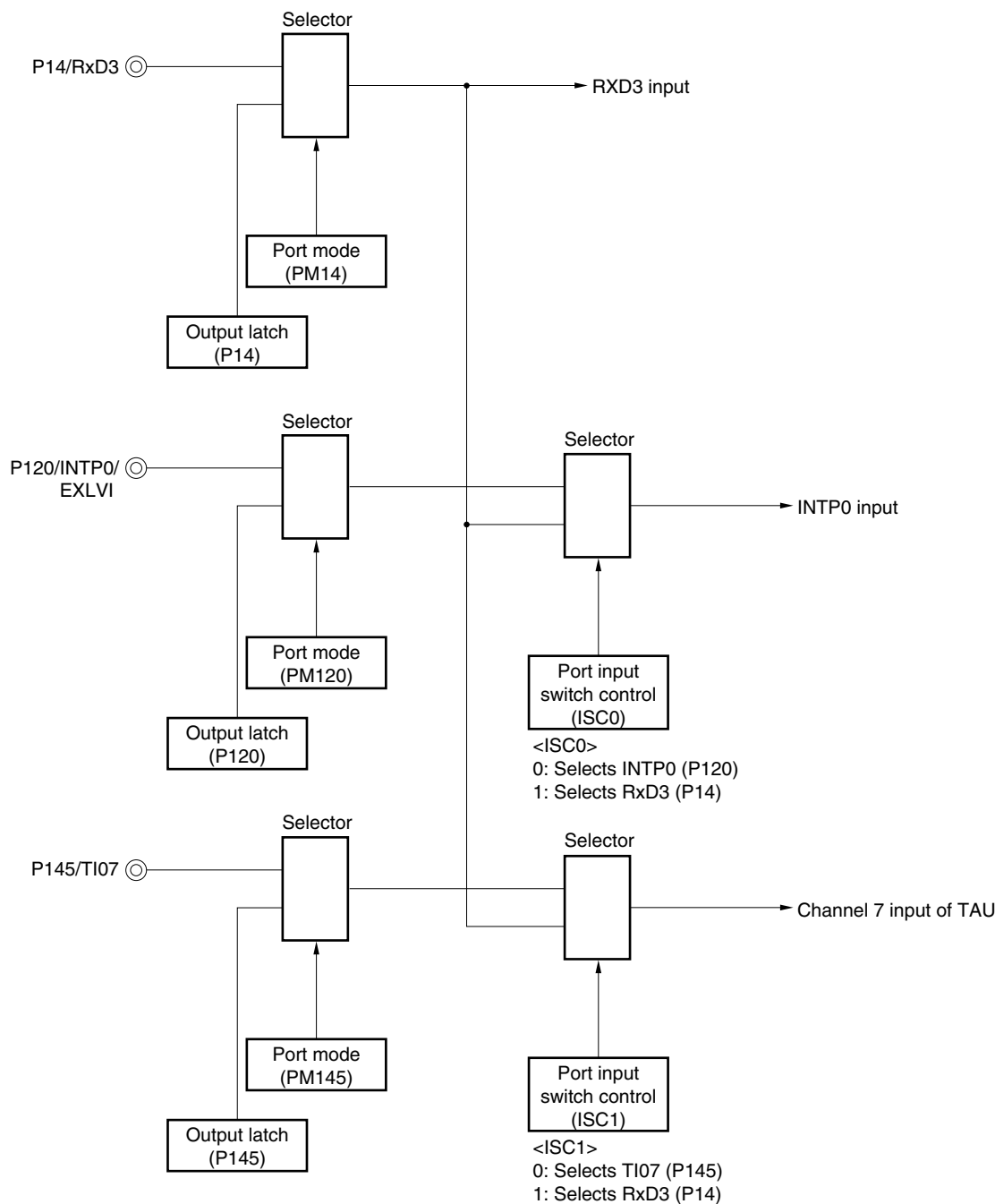
- <1> The wakeup signal is detected by detecting an interrupt edge (INTP0) on a pin. When the wakeup signal is detected, enable reception of UART3 (RXE13 = 1) and wait for SBF reception.
- <2> When the start bit of SBF is detected, reception is started and serial data is sequentially stored in the RXD3 register (= bits 7 to 0 of the serial data register 13 (SDR13)) at the set baud rate. When the stop bit is detected, the reception end interrupt request (INTSR3) is generated. When data of low levels of 11 bits or more is detected as SBF, it is judged that SBF reception has been correctly completed. If data of low levels of less than 11 bits is detected as SBF, it is judged that an SBF reception error has occurred, and the system returns to the SBF reception wait status.
- <3> When SBF reception has been correctly completed, start channel 7 of the timer array unit and measure the bit interval (pulse width) of the sync field (see 7.7.5 **Operation as input signal high-/low-level width measurement**).
- <4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART3 once and adjust (re-set) the baud rate.
- <5> The checksum field should be distinguished by software. In addition, processing to initialize UART3 after the checksum field is received and to wait for reception of SBF should also be performed by software.

Figure 13-88 shows the configuration of a port that manipulates reception of LIN.

The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (INTP0). The length of the sync field transmitted from the master can be measured by using the external event capture operation of the timer array unit (TAU) to calculate a baud-rate error.

By controlling switch of port input (ISC0/ISC1), the input source of port input (RxD3) for reception can be input to the external interrupt pin (INTP0) and timer array unit (TAU).

**Figure 13-88. Port Configuration for Manipulating Reception of LIN**



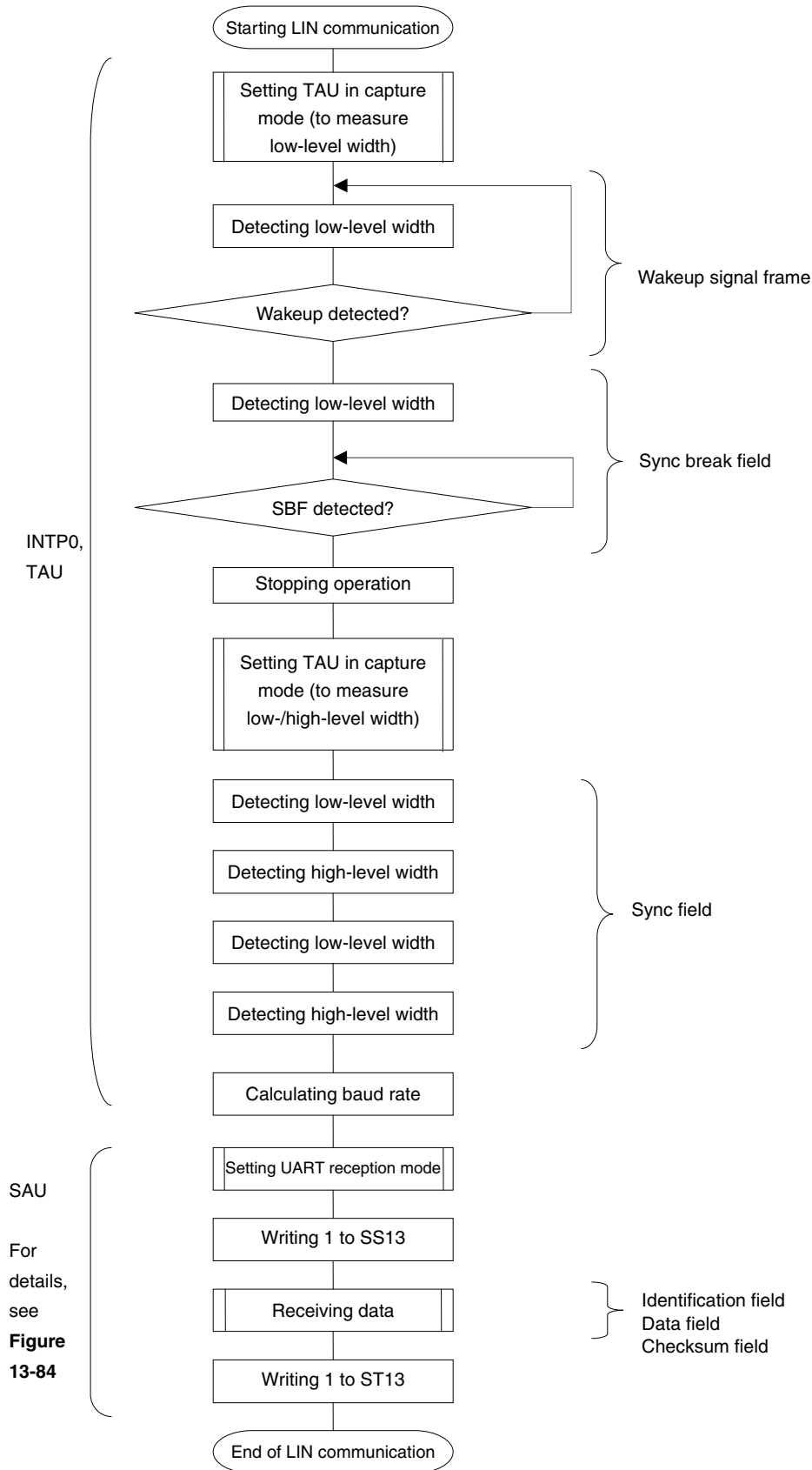
**Remark** ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 13-17**.)

The peripheral functions used for the LIN communication operation are as follows.

<Peripheral functions used>

- External interrupt (INTP0); Wakeup signal detection  
Usage: To detect an edge of the wakeup signal and the start of communication
- Channel 7 of timer array unit 0 (TAU0); Baud rate error detection  
Usage: To detect the length of the sync field (SF) and divide it by the number of bits in order to detect an error  
(The interval of the edge input to RxD3 is measured in the capture mode.)
- Channels 2 and 3 (UART3) of serial array unit 1 (SAU1)

Figure 13-89. Flowchart of LIN Reception



### 13.6.5 Calculating baud rate

#### (1) Baud rate calculation expression

The baud rate for UART (UART0, UART1, UART2, UART3) communication can be calculated by the following expressions.

$$\text{(Baud rate)} = \{\text{Operation clock (MCK) frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2 \text{ [bps]}$$

**Caution** Setting SDRmn [15:9] = (0000000B, 0000001B) is prohibited.

- Remarks**
1. When UART is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B) and therefore is 2 to 127.
  2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 13-3. Operating Clock Selection

SMRmn Register	SPSm Register								Operation Clock (MCK) <sup>Note1</sup>	
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	f <sub>CLK</sub> = 20 MHz
0	X	X	X	X	0	0	0	0	f <sub>CLK</sub>	20 MHz
	X	X	X	X	0	0	0	1	f <sub>CLK</sub> /2	10 MHz
	X	X	X	X	0	0	1	0	f <sub>CLK</sub> /2 <sup>2</sup>	5 MHz
	X	X	X	X	0	0	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	2.5 MHz
	X	X	X	X	0	1	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	1.25 MHz
	X	X	X	X	0	1	0	1	f <sub>CLK</sub> /2 <sup>5</sup>	625 kHz
	X	X	X	X	0	1	1	0	f <sub>CLK</sub> /2 <sup>6</sup>	313 kHz
	X	X	X	X	0	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	156 kHz
	X	X	X	X	1	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	78.1 kHz
	X	X	X	X	1	0	0	1	f <sub>CLK</sub> /2 <sup>9</sup>	39.1 kHz
	X	X	X	X	1	0	1	0	f <sub>CLK</sub> /2 <sup>10</sup>	19.5 kHz
	X	X	X	X	1	0	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0, INTTM03 if m = 1 <sup>Note2</sup>	
1	0	0	0	0	X	X	X	X	f <sub>CLK</sub>	20 MHz
	0	0	0	1	X	X	X	X	f <sub>CLK</sub> /2	10 MHz
	0	0	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>2</sup>	5 MHz
	0	0	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>3</sup>	2.5 MHz
	0	1	0	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>4</sup>	1.25 MHz
	0	1	0	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>5</sup>	625 kHz
	0	1	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>6</sup>	313 kHz
	0	1	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>7</sup>	156 kHz
	1	0	0	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>8</sup>	78.1 kHz
	1	0	0	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>9</sup>	39.1 kHz
	1	0	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>10</sup>	19.5 kHz
	1	0	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>11</sup>	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0, INTTM03 if m = 1 <sup>Note2</sup>	
Other than above									Setting prohibited	

**Notes 1.** When changing the clock selected for f<sub>CLK</sub> (by changing the system clock control register (CKC) value), do so after having stopped (STm = 000FH) the operation of the serial array unit (SAU). When selecting INTTM02 and INTTM03 for the operation clock, also stop the timer array unit 0 (TAU0) (TT0 = 00FFH).

**2.** SAU0 can be operated at a fixed division ratio of the subsystem clock, regardless of the f<sub>CLK</sub> frequency (main system clock, subsystem clock), by operating the interval timer for which f<sub>SUB</sub>/4 has been selected as the count clock (setting TIS02 (if m = 0) or TIS03 (if m = 1) of the TIS0 register to 1) and selecting INTTM02 and INTTM03 by using the SPSm register in channels 2 and 3 of TAU. When changing f<sub>CLK</sub>, however, SAU and TAU0 must be stopped as described in Note 1 above.

**Remarks 1.** X: Don't care

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

**(2) Baud rate error during transmission**

The baud rate error of UART (UART0, UART1, UART2, UART3) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$(\text{Baud rate error}) = (\text{Calculated baud rate value}) \div (\text{Target baud rate}) \times 100 - 100 [\%]$$

Here is an example of setting a UART baud rate at  $f_{\text{CLK}} = 20 \text{ MHz}$ .

UART Baud Rate (Target Baud Rate)	$f_{\text{CLK}} = 20 \text{ MHz}$			
	Operation Clock (MCK)	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{\text{CLK}}/2^9$	64	300.48 bps	+0.16 %
600 bps	$f_{\text{CLK}}/2^8$	64	600.96 bps	+0.16 %
1200 bps	$f_{\text{CLK}}/2^7$	64	1201.92 bps	+0.16 %
2400 bps	$f_{\text{CLK}}/2^6$	64	2403.85 bps	+0.16 %
4800 bps	$f_{\text{CLK}}/2^5$	64	4807.69 bps	+0.16 %
9600 bps	$f_{\text{CLK}}/2^4$	64	9615.38 bps	+0.16 %
19200 bps	$f_{\text{CLK}}/2^3$	64	19230.8 bps	+0.16 %
31250 bps	$f_{\text{CLK}}/2^3$	39	31250.0 bps	$\pm 0.0 \%$
38400 bps	$f_{\text{CLK}}/2^2$	64	38461.5 bps	+0.16 %
76800 bps	$f_{\text{CLK}}/2$	64	76923.1 bps	+0.16 %
153600 bps	$f_{\text{CLK}}$	64	153846 bps	+0.16 %
312500 bps	$f_{\text{CLK}}$	31	312500 bps	$\pm 0.0 \%$

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

**(3) Permissible baud rate range for reception**

The permissible baud rate range for reception during UART (UART0, UART1, UART2, UART3) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times \text{Nfr}}{2 \times k \times \text{Nfr} - k + 2} \times \text{Brate}$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (\text{Nfr} - 1)}{2 \times k \times \text{Nfr} - k - 2} \times \text{Brate}$$

Brate: Calculated baud rate value at the reception side (See **13.6.5 (1) Baud rate calculation expression.**)

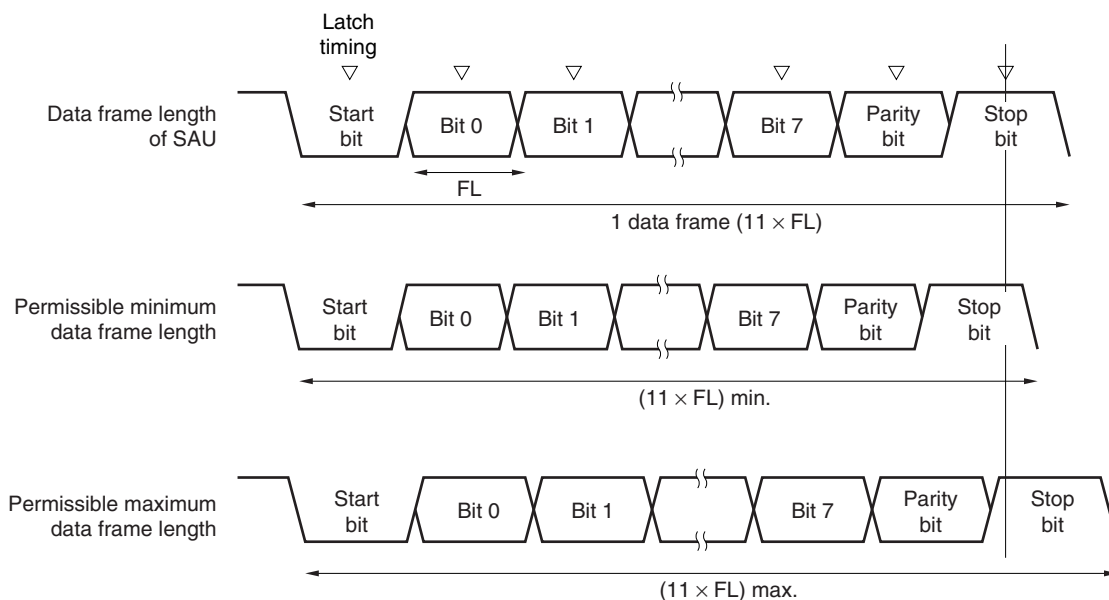
k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]

= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 1, 3)

**Figure 13-90. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)**



As shown in Figure 13-90, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of the serial data register mn (SDRmn) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.



### 13.6.6 Procedure for processing errors that occurred during UART (UART0, UART1, UART2, UART3) communication

The procedure for processing errors that occurred during UART (UART0, UART1, UART2, UART3) communication is described in Figures 13-91 and 13-92.

**Figure 13-91. Processing Procedure in Case of Parity Error or Overrun Error**

Software Manipulation	Hardware Status	Remark
Reads SDRmn register. →	The BFF0 = 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register. →	Error flag is cleared.	Only error generated at the point of reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

**Figure 13-92. Processing Procedure in Case of Framing Error**

Software Manipulation	Hardware Status	Remark
Reads SDRmn register. →	The BFF = 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register. →	Error flag is cleared.	Only error generated at the point of reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets STmn bit to 1. →	The SEmn = 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 SSmn bit to 1. →	The SEmn = 1, and channel n is enabled to operate.	

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

### 13.7 Operation of Simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master and does not have a wait detection function. Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function<sup>Note</sup> and ACK detection function
- Data length of 8 bits  
(When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- <R>
- Overrun error
  - Parity error (ACK error)

\* [Functions not supported by simplified I<sup>2</sup>C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection function

**Note** An ACK is not output when the last data is being received by writing 0 to the SOEmn (SOEm register) bit and stopping the output of serial communication data. See **13.7.3 (2) Processing flow** for details.

**Remarks 1.** To use the full-function I<sup>2</sup>C bus, see **CHAPTER 14 SERIAL INTERFACE IIC0**.

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

The channels supporting simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21) are channel 2 and 3 of SAU0 and channel 0 and 1 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I <sup>2</sup> C
0	0	CSI00	UART0	–
	1	CSI01		–
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	CSI21		IIC21
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

Simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21) performs the following four types of communication operations.

- Address field transmission (See **13.7.1**.)
- Data transmission (See **13.7.2**.)
- Data reception (See **13.7.3**.)
- Stop condition generation (See **13.7.4**.)

### 13.7.1 Address field transmission

Address field transmission is a transmission operation that first executes in I<sup>2</sup>C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Simplified I <sup>2</sup> C	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL10, SDA10 <sup>Note</sup>	SCL11, SDA11 <sup>Note</sup>	SCL20, SDA20 <sup>Note</sup>	SCL21, SDA21 <sup>Note</sup>
Interrupt	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	Parity error detection flag (PEFmn)			
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)			
Transfer rate	Max. $f_{MCK}/4$ [Hz] ( $SDRmn[15:9] = 1$ or more) $f_{MCK}$ : Operation clock (MCK) frequency of target channel However, the following condition must be satisfied in each mode of I <sup>2</sup> C. <ul style="list-style-type: none"> <li>• Max. 400 kHz (first mode)</li> <li>• Max. 100 kHz (standard mode)</li> </ul>			
Data level	Forward output (default: high level)			
Parity bit	No parity bit			
Stop bit	Appending 1 bit (for ACK reception timing)			
Data direction	MSB first			

**Note** To perform communication via simplified I<sup>2</sup>C, set the N-ch open-drain output ( $V_{DD}$  tolerance) mode (POM03, POM96, POM126, POM143 = 1) for the port output mode registers (POM0, POM9, POM12, POM14) (see **4.3 Registers Controlling Port Function** for details). When communicating with an external device with a different potential, set the N-ch open-drain output ( $V_{DD}$  tolerance) mode (POM04, POM95, POM125, POM142 = 1) also for the clock input/output pins (SCL10, SCL11, SCL20, SCL21) (see **4.4.4 Connecting to external device with different potential (2.5 V, 3 V)** for details).

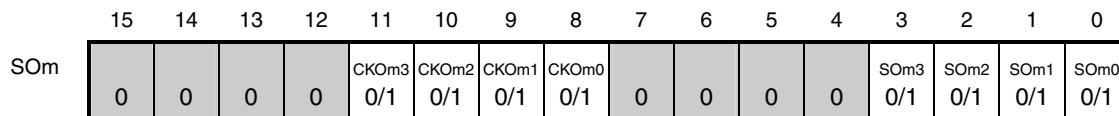
**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

<R>

(1) Register setting

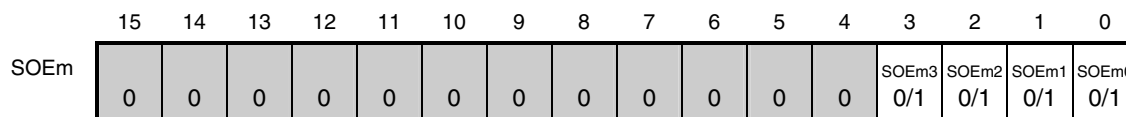
Figure 13-93. Example of Contents of Registers for Address Field Transmission of Simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21)

(a) Serial output register m (SOM) ... Sets only the bits of the target channel.



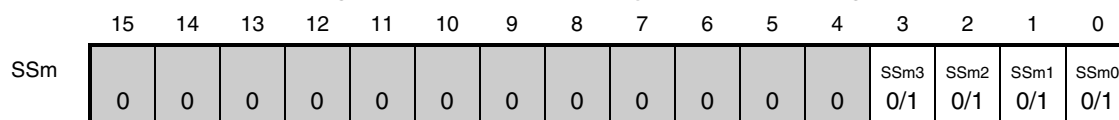
Start condition is generated by manipulating the SOMn bit.

(b) Serial output enable register m (SOEm) ... Sets only the bits of the target channel.

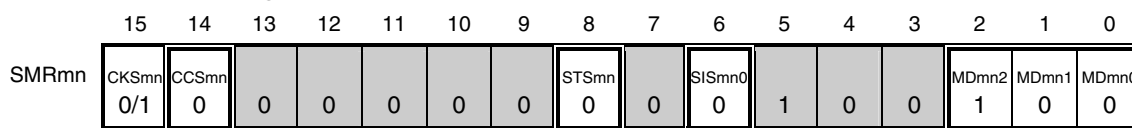


SOEmn = 0 until the start condition is generated, and SOEmn = 1 after generation.

(c) Serial channel start register m (SSm) ... Sets only the bits of the target channel is 1.

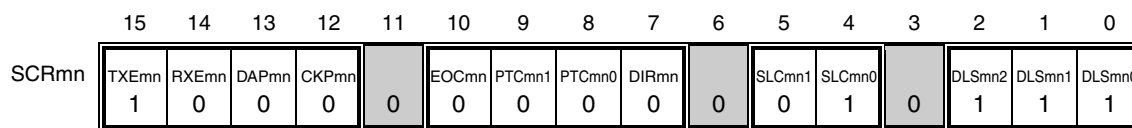


(d) Serial mode register mn (SMRmn)



Interrupt sources of channel n  
0: Transfer end interrupt

(e) Serial communication operation setting register mn (SCRmn)



Setting of parity bit  
00B: No parity

Setting of stop bit  
01B: Appending 1 bit

(f) Serial data register mn (SDRmn) (lower 8 bits: SIO<sub>r</sub>)



SIO<sub>r</sub>

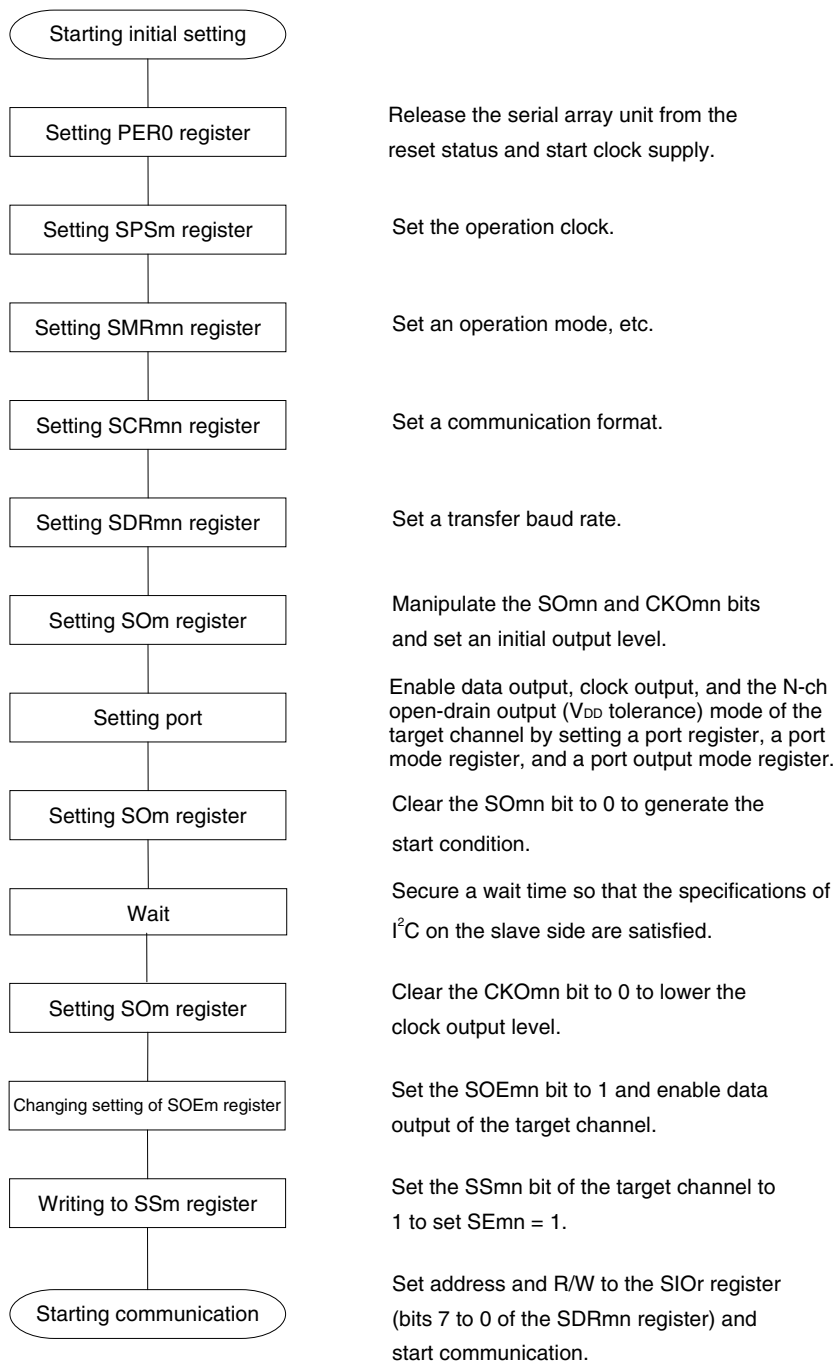
**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 10, 11, 20, 21)

: Setting is fixed in the IIC mode, : Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

## (2) Operation procedure

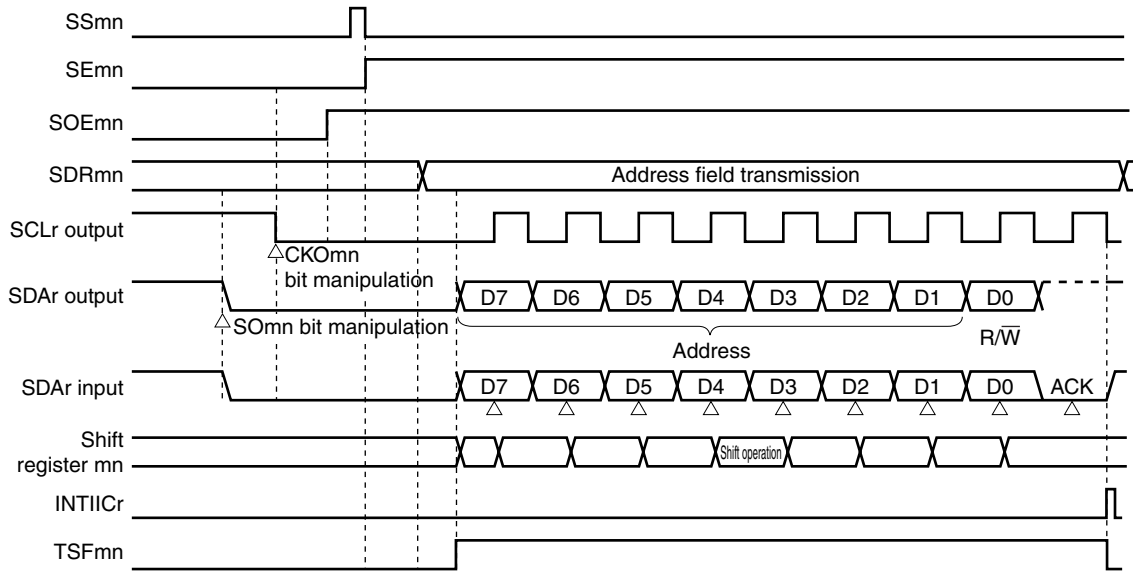
Figure 13-94. Initial Setting Procedure for Address Field Transmission



**Caution** After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

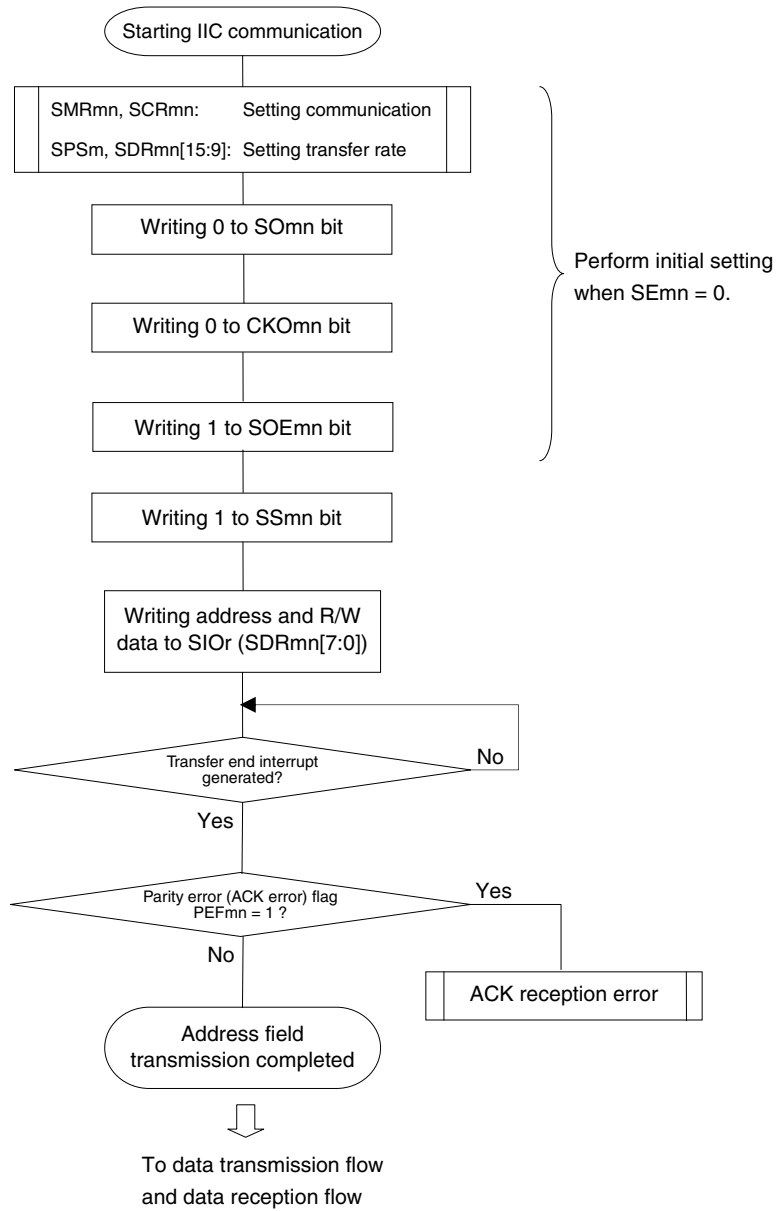
(3) Processing flow

Figure 13-95. Timing Chart of Address Field Transmission



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 10, 11, 20, 21)

Figure 13-96. Flowchart of Address Field Transmission



### 13.7.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I <sup>2</sup> C	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL10, SDA10 <sup>Note</sup>	SCL11, SDA11 <sup>Note</sup>	SCL20, SDA20 <sup>Note</sup>	SCL21, SDA21 <sup>Note</sup>
Interrupt	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	Parity error detection flag (PEFmn)			
Transfer data length	8 bits			
<R> Transfer rate	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) $f_{MCK}$ : Operation clock (MCK) frequency of target channel However, the following condition must be satisfied in each mode of I <sup>2</sup> C. <ul style="list-style-type: none"> <li>• Max. 400 kHz (first mode)</li> <li>• Max. 100 kHz (standard mode)</li> </ul>			
Data level	Forward output (default: high level)			
Parity bit	No parity bit			
Stop bit	Appending 1 bit (for ACK reception timing)			
Data direction	MSB first			

**Note** To perform communication via simplified I<sup>2</sup>C, set the N-ch open-drain output ( $V_{DD}$  tolerance) mode (POM03, POM96, POM126, POM143 = 1) for the port output mode registers (POM0, POM9, POM12, POM14) (see **4.3 Registers Controlling Port Function** for details). When communicating with an external device with a different potential, set the N-ch open-drain output ( $V_{DD}$  tolerance) mode (POM04, POM95, POM125, POM142 = 1) also for the clock input/output pins (SCL10, SCL11, SCL20, SCL21) (see **4.4.4 Connecting to external device with different potential (2.5 V, 3 V)** for details).

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)



(1) Register setting

Figure 13-97. Example of Contents of Registers for Data Transmission of Simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21)

(a) Serial output register m (SOm) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm					CKOm3	CKOm2	CKOm1	CKOm0					SOm3	SOm2	SOm1	SOm0
	0	0	0	0	×	0/1 <sup>Note</sup>	×	0/1 <sup>Note</sup>	0	0	0	0	×	0/1 <sup>Note</sup>	×	0/1 <sup>Note</sup>

(b) Serial output enable register m (SOEm) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

(c) Serial channel start register m (SSm) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

(d) Serial mode register mn (SMRmn) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(e) Serial communication operation setting register mn (SCRmn) ... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0		DLSmn2	DLSmn1	DLSmn0
	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

(f) Serial data register mn (SDRmn) (lower 8 bits: SIO<sub>r</sub>)

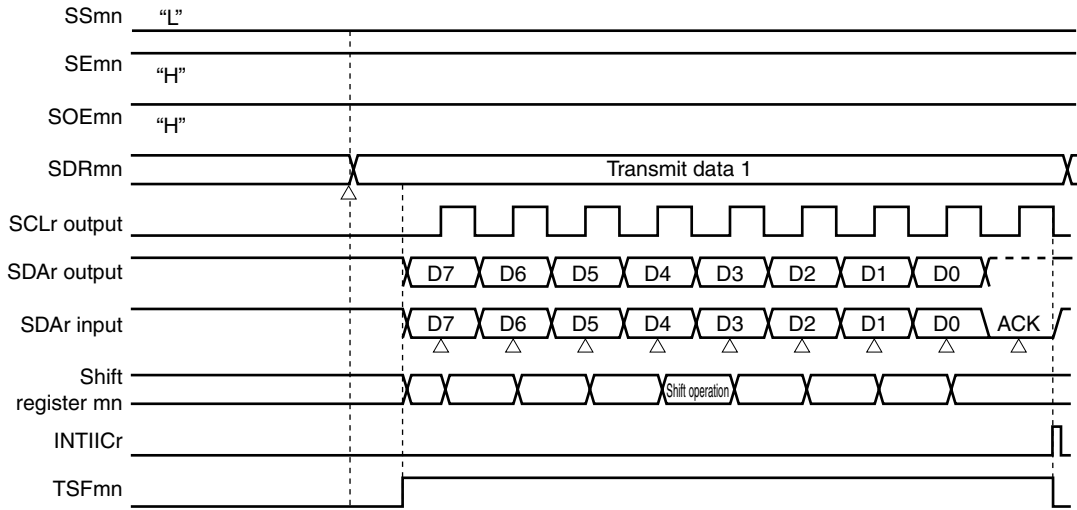
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting								0	Transmit data setting						
	SIO <sub>r</sub>															

**Note** The value varies depending on the communication data during communication operation.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 10, 11, 20, 21)  
: Setting is fixed in the IIC mode, : Setting disabled (set to the initial value)  
 ×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)  
 0/1: Set to 0 or 1 depending on the usage of the user

(2) Processing flow

Figure 13-98. Timing Chart of Data Transmission



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 10, 11, 20, 21)

Figure 13-99. Flowchart of Data Transmission



## 13.7.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I <sup>2</sup> C	IIC10	IIC11	IIC20	IIC21
Target channel	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1
Pins used	SCL10, SDA10 <sup>Note</sup>	SCL11, SDA11 <sup>Note</sup>	SCL20, SDA20 <sup>Note</sup>	SCL21, SDA21 <sup>Note</sup>
Interrupt	INTIIC10	INTIIC11	INTIIC20	INTIIC21
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	Overflow error detection flag (OVFmn) only			
Transfer data length	8 bits			
Transfer rate	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) $f_{MCK}$ : Operation clock (MCK) frequency of target channel However, the following condition must be satisfied in each mode of I <sup>2</sup> C. <ul style="list-style-type: none"> <li>• Max. 400 kHz (first mode)</li> <li>• Max. 100 kHz (standard mode)</li> </ul>			
Data level	Forward output (default: high level)			
Parity bit	No parity bit			
Stop bit	Appending 1 bit (ACK transmission)			
Data direction	MSB first			

**Note** To perform communication via simplified I<sup>2</sup>C, set the N-ch open-drain output (V<sub>DD</sub> tolerance) mode (POM03, POM96, POM126, POM143 = 1) for the port output mode registers (POM0, POM9, POM12, POM14) (see **4.3 Registers Controlling Port Function** for details). When communicating with an external device with a different potential, set the N-ch open-drain output (V<sub>DD</sub> tolerance) mode (POM04, POM95, POM125, POM142 = 1) also for the clock input/output pins (SCL10, SCL11, SCL20, SCL21) (see **4.4.4 Connecting to external device with different potential (2.5 V, 3 V)** for details).

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

(1) Register setting

Figure 13-100. Example of Contents of Registers for Data Reception of Simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21)

(a) Serial output register m (SOM) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOM					CKOm3	CKOm2	CKOm1	CKOm0					SOM3	SOM2	SOM1	SOM0
	0	0	0	0	×	0/1 <sup>Note</sup>	×	0/1 <sup>Note</sup>	0	0	0	0	×	0/1 <sup>Note</sup>	×	0/1 <sup>Note</sup>

(b) Serial output enable register m (SOEm) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

(c) Serial channel start register m (SSm) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

(d) Serial mode register mn (SMRmn) ... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SlSmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(e) Serial communication operation setting register mn (SCRmn) ... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0		DLSmn2	DLSmn1	DLSmn0
	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1

(f) Serial data register mn (SDRmn) (lower 8 bits: SIO<sub>r</sub>)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting								0	Dummy transmit data setting (FFH)						
	SIO <sub>r</sub>															

**Note** The value varies depending on the communication data during communication operation.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 10, 11, 20, 21)

□: Setting is fixed in the IIC mode, ■: Setting disabled (set to the initial value)

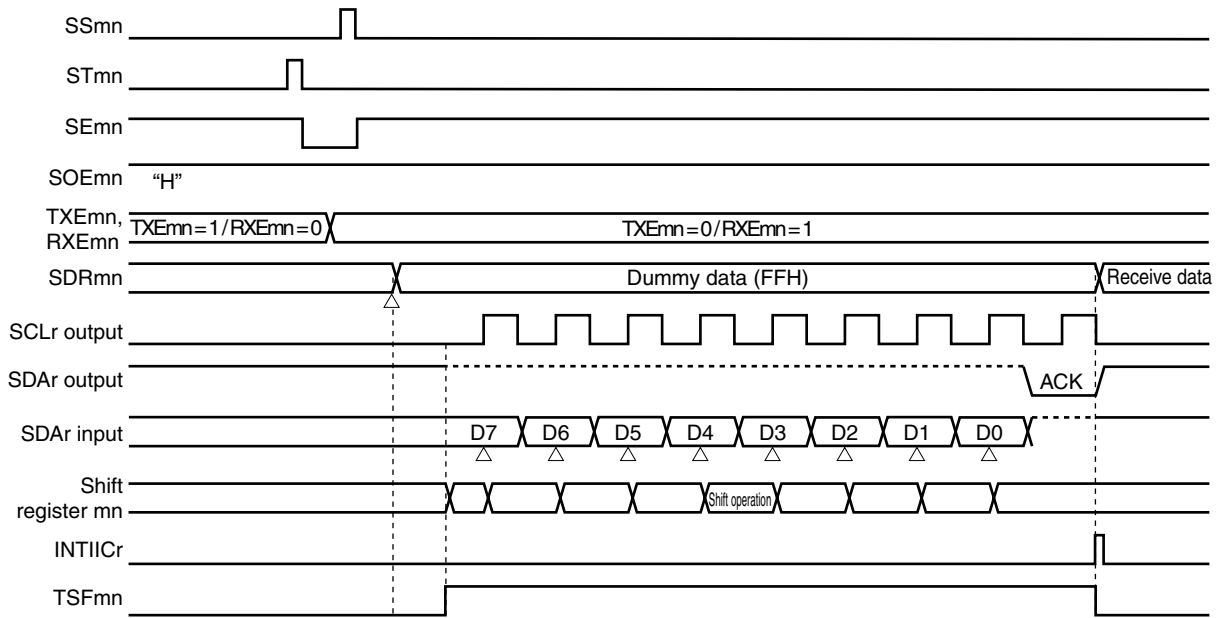
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

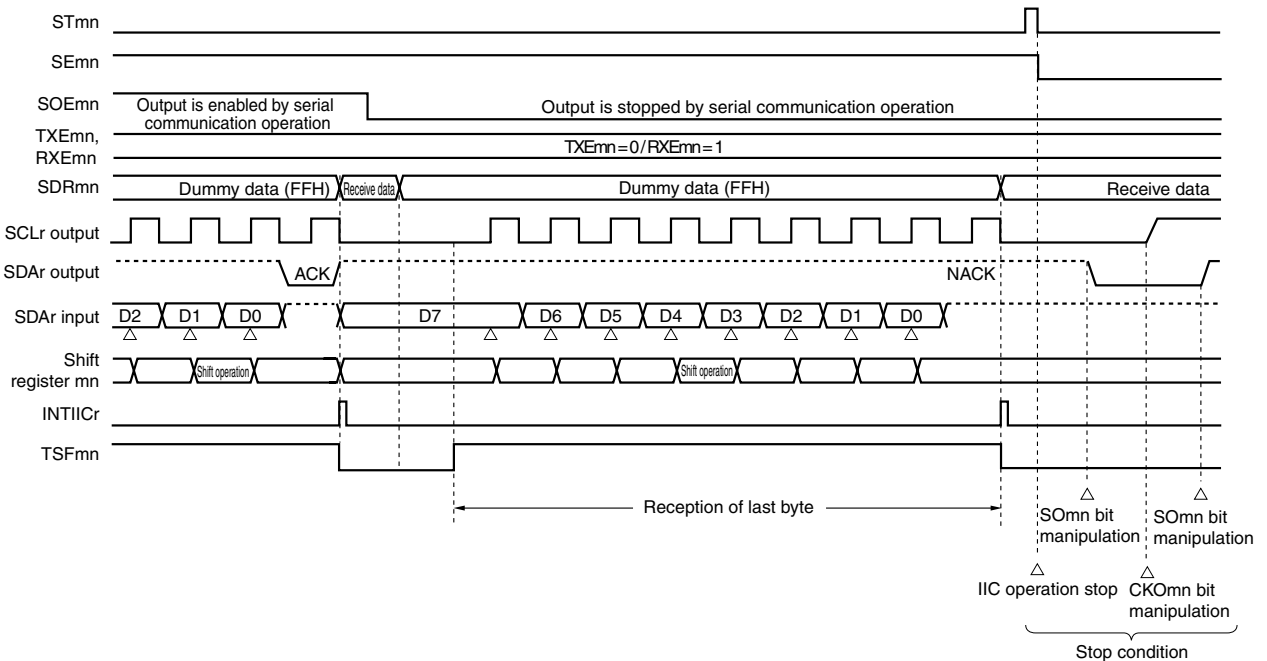
(2) Processing flow

Figure 13-101. Timing Chart of Data Reception

(a) When starting data reception

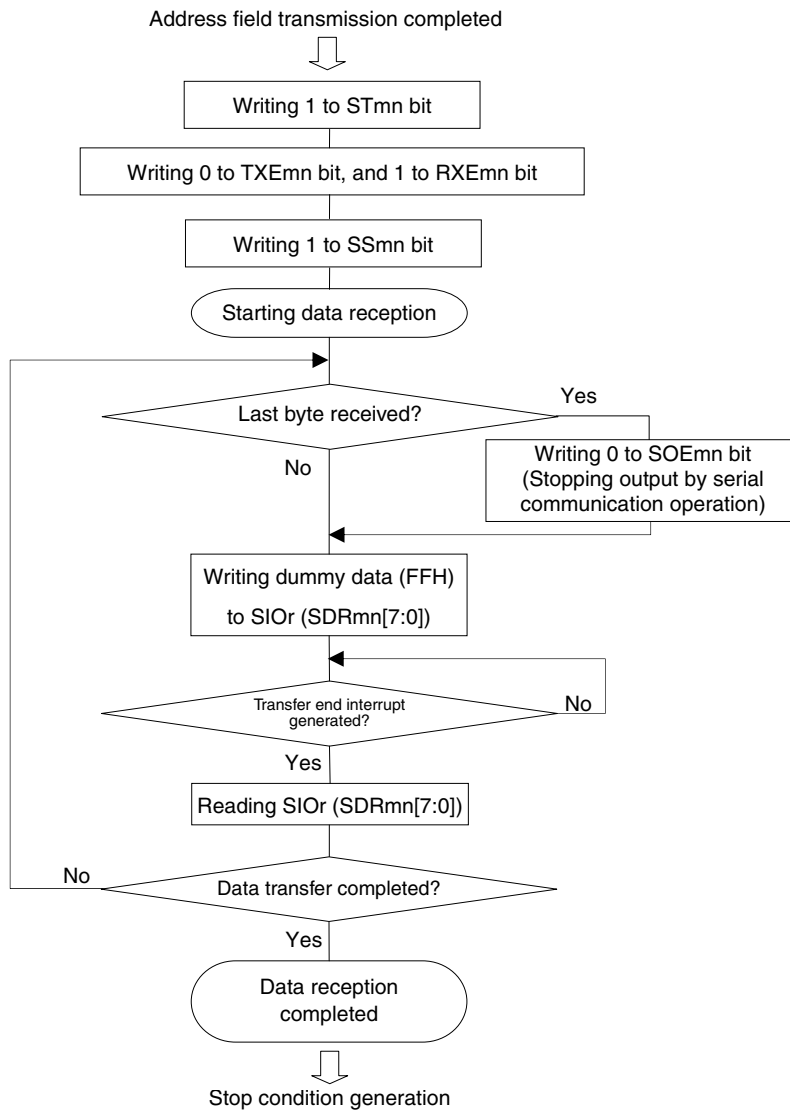


(b) When receiving last data



**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 10, 11, 20, 21)

Figure 13-102. Flowchart of Data Reception



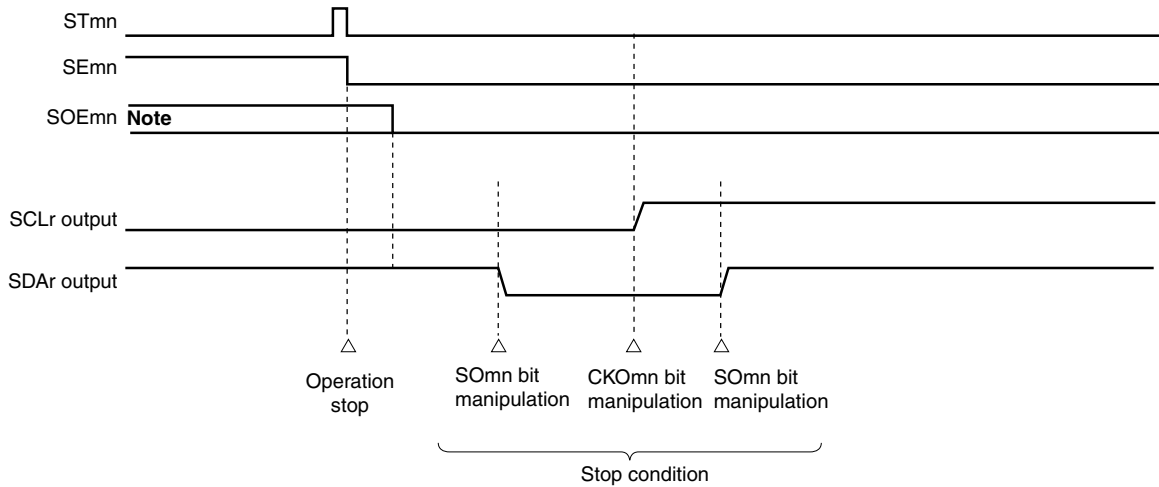
**Caution** ACK is not output when the last data is received (NACK). Communication is then completed by setting “1” to the STmn bit to stop operation and generating a stop condition.

13.7.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

(1) Processing flow

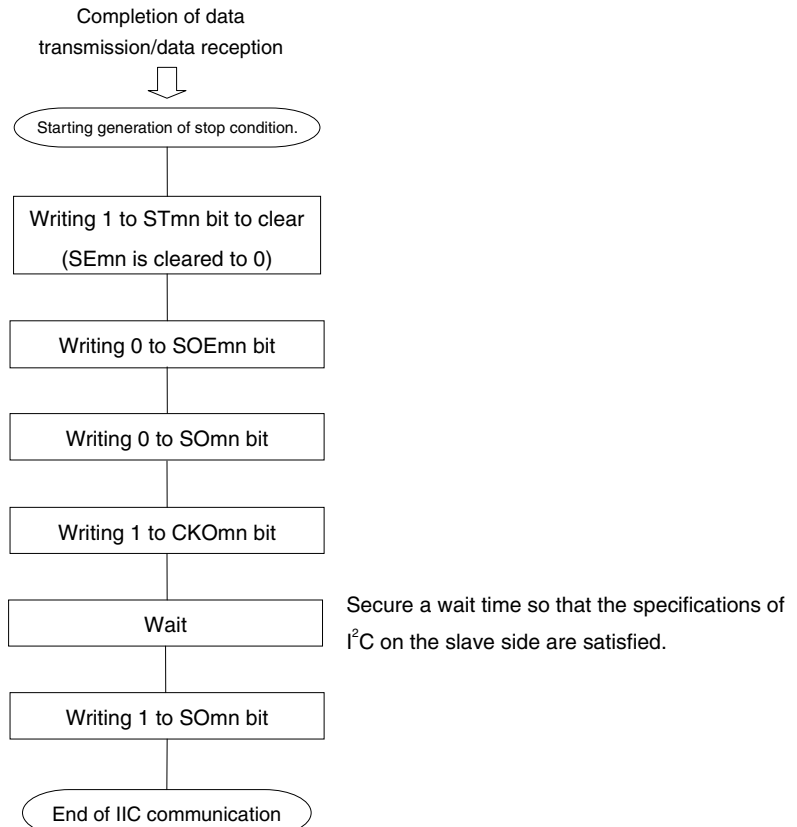
Figure 13-103. Timing Chart of Stop Condition Generation



**Note** During the receive operation, the SOEmn bit is set to 0 before receiving the last data.

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 10, 11, 20, 21)

Figure 13-104. Flowchart of Stop Condition Generation



### 13.7.5 Calculating transfer rate

The transfer rate for simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21) communication can be calculated by the following expressions.

$$(\text{Transfer rate}) = \{\text{Operation clock (MCK) frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2$$

<R>        **Caution** Setting SDRmn[15:9] = 0000000B is prohibited. Setting SDRmn[15:9] = 0000001B or more.

<R>        **Remarks** 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000001B to 1111111B) and therefore is 1 to 127.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).



Table 13-4. Operating Clock Selection

SMRmn Register	SPSm Register								Operation Clock (MCK) <sup>Note1</sup>	
	CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00	f <sub>CLK</sub> = 20 MHz
0	X	X	X	X	0	0	0	0	f <sub>CLK</sub>	20 MHz
	X	X	X	X	0	0	0	1	f <sub>CLK</sub> /2	10 MHz
	X	X	X	X	0	0	1	0	f <sub>CLK</sub> /2 <sup>2</sup>	5 MHz
	X	X	X	X	0	0	1	1	f <sub>CLK</sub> /2 <sup>3</sup>	2.5 MHz
	X	X	X	X	0	1	0	0	f <sub>CLK</sub> /2 <sup>4</sup>	1.25 MHz
	X	X	X	X	0	1	0	1	f <sub>CLK</sub> /2 <sup>5</sup>	625 kHz
	X	X	X	X	0	1	1	0	f <sub>CLK</sub> /2 <sup>6</sup>	313 kHz
	X	X	X	X	0	1	1	1	f <sub>CLK</sub> /2 <sup>7</sup>	156 kHz
	X	X	X	X	1	0	0	0	f <sub>CLK</sub> /2 <sup>8</sup>	78.1 kHz
	X	X	X	X	1	0	0	1	f <sub>CLK</sub> /2 <sup>9</sup>	39.1 kHz
	X	X	X	X	1	0	1	0	f <sub>CLK</sub> /2 <sup>10</sup>	19.5 kHz
	X	X	X	X	1	0	1	1	f <sub>CLK</sub> /2 <sup>11</sup>	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0, INTTM03 if m = 1 <sup>Note2</sup>	
1	0	0	0	0	X	X	X	X	f <sub>CLK</sub>	20 MHz
	0	0	0	1	X	X	X	X	f <sub>CLK</sub> /2	10 MHz
	0	0	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>2</sup>	5 MHz
	0	0	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>3</sup>	2.5 MHz
	0	1	0	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>4</sup>	1.25 MHz
	0	1	0	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>5</sup>	625 kHz
	0	1	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>6</sup>	313 kHz
	0	1	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>7</sup>	156 kHz
	1	0	0	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>8</sup>	78.1 kHz
	1	0	0	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>9</sup>	39.1 kHz
	1	0	1	0	X	X	X	X	f <sub>CLK</sub> /2 <sup>10</sup>	19.5 kHz
	1	0	1	1	X	X	X	X	f <sub>CLK</sub> /2 <sup>11</sup>	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0, INTTM03 if m = 1 <sup>Note2</sup>	
Other than above									Setting prohibited	

**Notes 1.** When changing the clock selected for f<sub>CLK</sub> (by changing the system clock control register (CKC) value), do so after having stopped (STm = 000FH) the operation of the serial array unit (SAU). When selecting INTTM02 and INTTM03 for the operation clock, also stop the timer array unit 0 (TAU0) (TT0 = 00FFH).

**2.** SAU0 can be operated at a fixed division ratio of the subsystem clock, regardless of the f<sub>CLK</sub> frequency (main system clock, subsystem clock), by operating the interval timer for which f<sub>SUB</sub>/4 has been selected as the count clock (setting TIS02 (if m = 0) or TIS03 (if m = 1) of the TIS0 register to 1) and selecting INTTM02 and INTTM03 by using the SPSm register in channels 2 and 3 of TAU. When changing f<sub>CLK</sub>, however, SAU and TAU0 must be stopped as described in Note 1 above.

**Remarks 1.** X: Don't care

**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Here is an example of setting an IIC transfer rate where  $MCK = f_{CLK} = 20\text{ MHz}$ .

IIC Transfer Mode (Desired Transfer Rate)	$f_{CLK} = 20\text{ MHz}$			
	Operation Clock (MCK)	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}$	99	100 kHz	0.0%
400 kHz	$f_{CLK}$	24	400 kHz	0.0%

### 13.7.6 Procedure for processing errors that occurred during simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21) communication

The procedure for processing errors that occurred during simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21) communication is described in Figures 13-105 and 13-106.

&lt;R&gt;

**Figure 13-105. Processing Procedure in Case of Parity Error or Overrun Error**

Software Manipulation	Hardware Status	Remark
Reads SDRmn register. —————>	▶ BFF = 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register. —————>	▶ Error flag is cleared.	Only error generated at the point of reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

**Figure 13-106. Processing Procedure in Case of Parity Error (ACK error) in Simplified I<sup>2</sup>C Mode**

Software Manipulation	Hardware Status	Remark
Reads SDRmn register. —————>	▶ BFF = 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register. —————>	▶ Error flag is cleared.	Only error generated at the point of reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets STmn bit to 1. —————>	▶ SEMn = 0, and channel n stops operation.	Slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates stop condition.		
Creates start condition.		
Sets SSmn bit to 1. —————>	▶ SEMn = 1, and channel n is enabled to operate.	

**Remark** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 02, 03, 10, 11

### 13.8 Relationship Between Register Settings and Pins

Tables 13-5 to 13-12 show the relationship between register settings and pins for each channel of serial array units 0 and 1.

**Table 13-5. Relationship between register settings and pins (Channel 0 of unit 0: CSI00, UART0 transmission)**

SE 00 Note1	MD 002	MD 001	SOE 00	SO 00	CKO 00	TXE 00	RXE 00	PM1 0	P10	PM 11 Note2	P11 Note2	PM 12	P12	Operation mode	Pin Function			
															SCK00/ EX24/P10	SI00/EX25/ RxD0/P11 Note2	SO00/EX26/ TxD0/P12	
0	0	0	0	1	1	0	0	×	×	×	×	×	×	Operation stop mode	EX24/P10	EX25/P11	EX26/P12	
	0	1														EX25/P11/ RxD0		
1	0	0	0	1	1	0	1	1	×	1	×	×	×	Slave CSI00 reception	SCK00 (input)	SI00	P12	
			1	0/1 Note4	1	1	0	1	×	×	×	0	1	Slave CSI00 transmission	SCK00 (input)	P11	SO00	
			1	0/1 Note4	1	1	1	1	×	1	×	0	1	Slave CSI00 transmission/reception	SCK00 (input)	SI00	SO00	
			0	1	0/1 Note4	0	1	0	1	1	×	×	×	×	Master CSI00 reception	SCK00 (output)	SI00	P12
			1	0/1 Note4	0/1 Note4	1	0	0	1	×	×	×	0	1	Master CSI00 transmission	SCK00 (output)	P11	SO00
			1	0/1 Note4	0/1 Note4	1	1	0	1	1	×	×	0	1	Master CSI00 transmission/reception	SCK00 (output)	SI00	SO00
0	1	1	0/1 Note4	1	1	0	×	×	×	×	0	1	UART0 transmission Note5	P10	EX25/P11/ RxD0	TxD0		

- Notes**
1. The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.
  2. When channel 1 of unit 0 is set to UART0 reception, this pin becomes an RxD0 function pin (refer to **Table 13-6**). In this case, operation stop mode or UART0 transmission must be selected for channel 0 of unit 0.
  3. This pin can be set as a port function pin.
  4. This is 0 or 1, depending on the communication operation. For details, refer to **13.3 (12) Serial output register m (SOM)**.
  5. When using UART0 transmission and reception in a pair, set channel 1 of unit 0 to UART0 reception (refer to **Table 13-6**).

**Remark** X: Don't care

**Table 13-6. Relationship between register settings and pins (Channel 1 of unit 0: CSI01, UART0 reception)**

SE 01 Note1	MD 012	MD 011	SOE 01	SO01	CKO 01	TXE 01	RXE 01	PM 43	P43	PM44	P44	PM 45	P45	PM 11 Note2	P11 Note2	Operation mode	Pin Function					
																	SCK01/ P43	SI01/P44	SO01/ P45	SI00/EX25/ P11 Note2		
0	0	0	0	1	1	0	0	×	×	×	×	×	×	×	×	Operation stop mode	P43	P44	P45	SI00/EX25/ P11		
																					0	1
1	0	0	0	1	1	0	1	1	×	1	×	×	×	×	×	Slave CSI01 reception	SCK01 (input)	SI01	P45	SI00/EX25/ P11		
				1	0/1 Note4	1	1	0	1	×	×	×	0	1	×	×	Slave CSI01 transmission	SCK01 (input)	P44	SO01	SI00/EX25/ P11	
				1	0/1 Note4	1	1	1	1	×	1	×	0	1	×	×	Slave CSI01 transmission /reception	SCK01 (input)	SI01	SO01	SI00/EX25/ P11	
				0	1	0/1 Note4	0	1	0	1	1	×	×	×	×	×	Master CSI01 reception	SCK01 (output)	SI01	P45	SI00/EX25/ P11	
				1	0/1 Note4	0/1 Note4	1	0	0	1	×	×	×	0	1	×	×	Master CSI01 transmission	SCK01 (output)	P44	SO01	SI00/EX25/ P11
				1	0/1 Note4	0/1 Note4	1	1	0	1	1	×	×	0	1	×	×	Master CSI01 transmission /reception	SCK01 (output)	SI01	SO01	SI00/EX25/ P11
0	1	0	1	1	0	1	×	×	×	×	×	×	1	×	UART0 reception Notes 5, 6	P43	P44	P45	RxD0			

- Notes**
- The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.
  - When channel 1 of unit 0 is set to UART0 reception, this pin becomes an RxD0 function pin. In this case, set channel 0 of unit 0 to operation stop mode or UART0 transmission (refer to **Table 13-5**).  
When channel 0 of unit 0 is set to CSI00, this pin cannot be used as an RxD0 function pin. In this case, set channel 1 of unit 0 to operation stop mode or CSI01.
  - This pin can be set as a port function pin.
  - This is 0 or 1, depending on the communication operation. For details, refer to **13.3 (12) Serial output register m (S0m)**.
  - When using UART0 transmission and reception in a pair, set channel 0 of unit 0 to UART0 transmission (refer to **Table 13-5**).
  - The SMR00 register of channel 0 of unit 0 must also be set during UART0 reception. For details, refer to **13.6.2 (1) Register setting**.

**Remark** X: Don't care

**Table 13-7. Relationship between register settings and pins  
(Channel 2 of unit 0: CSI10, UART1 transmission, IIC10)**

SE 02 Note1	MD 022	MD 021	SOE 02	SO 02	CKO 02	TXE 02	RXE 02	PM 04	P04	PM03 Note2	P03 Note2	PM02	P02	Operation mode	Pin Function			
															SCK10/ SCL10/P04	SI10/SDA10/ RxD1/P03 Note2	SO10/ TxD1/P02	
0	0	0	0	1	1	0	0	×	×	×	×	×	×	Operation stop mode	P04	P03	P02	
	0	1														P03/RxD1		
	1	0														P03		
1	0	0	0	1	1	0	1	1	×	1	×	×	×	Slave CSI10 reception	SCK10 (input)	SI10	P02	
			1	0/1 Note4	1	1	0	1	×	×	×	0	1	Slave CSI10 transmission	SCK10 (input)	P03	SO10	
			1	0/1 Note4	1	1	1	1	×	1	×	0	1	Slave CSI10 transmission /reception	SCK10 (input)	SI10	SO10	
			0	1	0/1 Note4	0	1	0	1	1	×	×	×	×	Master CSI10 reception	SCK10 (output)	SI10	P02
			1	0/1 Note4	0/1 Note4	1	0	0	1	×	×	×	0	1	Master CSI10 transmission	SCK10 (output)	P03	SO10
			1	0/1 Note4	0/1 Note4	1	1	0	1	1	×	×	0	1	Master CSI10 transmission /reception	SCK10 (output)	SI10	SO10
	0	1	1	0/1 Note4	1	1	0	×	×	×	×	0	1	UART1 transmission Note5	P04	P03/RxD1	TxD1	
0	1	0	0	0/1 Note6	0/1 Note6	0	0	0	1	0	1	×	×	IIC10 start condition	SCL10	SDA10	P02	
						1	0											
						0	1											
1			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	×	×	IIC10 address field transmission	SCL10	SDA10	P02	
			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	×	×	IIC10 data transmission	SCL10	SDA10	P02	
			1	0/1 Note4	0/1 Note4	0	1	0	1	0	1	×	×	IIC10 data reception	SCL10	SDA10	P02	
0			0	0/1 Note7	0/1 Note7	0	0	0	1	0	1	×	×	IIC10 stop condition	SCL10	SDA10	P02	
						1	0											
						0	1											

- Notes 1.** The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.
- When channel 3 of unit 0 is set to UART1 reception, this pin becomes an RxD1 function pin (refer to **Table 13-8**). In this case, operation stop mode or UART1 transmission must be selected for channel 2 of unit 0.
  - This pin can be set as a port function pin.
  - This is 0 or 1, depending on the communication operation. For details, refer to **13.3 (12) Serial output register m (SOM)**.
  - When using UART1 transmission and reception in a pair, set channel 3 of unit 0 to UART1 reception (refer to **Table 13-8**).
  - Set the CKO02 bit to 1 before a start condition is generated. Clear the SO02 bit from 1 to 0 when the start condition is generated.
  - Set the CKO02 bit to 1 before a stop condition is generated. Clear the SO02 bit from 0 to 1 when the stop condition is generated.

**Remark** X: Don't care

Table 13-8. Relationship between register settings and pins (Channel 3 of unit 0: CSI11, UART1 reception, IIC11)

SE 03 Note1	MD 032	MD 031	SOE 03	SO03	CKO 03	TXE 03	RXE 03	PM 95	P95	PM96	P96	PM 97	P97	PM 03 Note2	P03 Note2	Operation mode	Pin Function																							
																	SCK11/ SCL11/ P95	SI11/ SDA11/ P96	SO11/ P97	SI10/ SDA10/ RxD1/ P03 Note2																				
0	0	0	0	1	1	0	0	×	×	×	×	×	×	×	×	Operation stop mode	P95	P96	P97	SI00/ SDA10/ P03 Note2																				
																					0	1																		
																					1	0																		
1	0	0	0	1	1	0	1	1	×	1	×	×	×	×	×	Slave CSI11 reception	SCK11 (input)	SI11	P97	SI00/ SDA10/ P03 Note2																				
																					1	0/1 Note4	1	1	0	1	×	×	×	0	1	×	×	Slave CSI11 transmission	SCK11 (input)	P96	SO11	SI00/ SDA10/ P03 Note2		
																					1	0/1 Note4	1	1	1	1	×	1	×	0	1	×	×	Slave CSI11 transmission /reception	SCK11 (input)	SI11	SO11	SI00/ SDA10/ P03 Note2		
																					0	1	0/1 Note4	0	1	0	1	1	×	×	×	×	×	Master CSI11 reception	SCK11 (output)	SI11	P97	SI00/ SDA10/ P03 Note2		
																					1	0/1 Note4	0/1 Note4	1	0	0	1	×	×	×	0	1	×	×	Master CSI11 transmission	SCK11 (output)	P96	SO11	SI00/ SDA10/ P03 Note2,5/ P11	
																					1	0/1 Note4	0/1 Note4	1	1	0	1	1	×	×	×	0	1	×	×	Master CSI11 transmission /reception	SCK11 (output)	SI11	SO11	SI00/ SDA10/ P03 Note2
																					0	1	0	1	1	0	1	×	×	×	×	×	×	1	×	UART1 reception Notes 5, 6	P95	P96	P97	RxD1
0	1	0	0	0/1 Note7	0/1 Note7	0	0	0	1	0	1	×	×	×	×	IIC11 start condition	SCL11	SDA11	P97	SI00/ SDA10/ P03 Note2																				
																					1	0																		
																					0	1																		
1			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	×	×	×	×	IIC11 address field transmission	SCL11	SDA11	P97	SI00/ SDA10/ P03 Note2																				
																					1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	×	×	×	×	IIC11 data transmission	SCL11	SDA11	P97	SI00/ SDA10/ P03 Note2		
																					1	0/1 Note4	0/1 Note4	0	1	0	1	0	1	×	×	×	×	IIC11 data reception	SCL11	SDA11	P97	SI00/ SDA10/ P03 Note2		
0			0	0/1 Note8	0/1 Note8	0	0	0	1	0	1	×	×	×	×	IIC11 stop condition	SCL11	SDA11	P97	SI00/ SDA10/ P03 Note2																				

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

- Notes**
1. The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.
  2. When channel 3 of unit 0 is set to UART1 reception, this pin becomes an RxD1 function pin. In this case, set channel 2 of unit 0 to operation stop mode or UART1 transmission (refer to **Table 13-7**).  
When channel 2 of unit 0 is set to CSI10 or IIC10, this pin cannot be used as an RxD1 function pin. In this case, set channel 3 of unit 0 to operation stop mode or CSI11 or IIC11.
  3. This pin can be set as a port function pin.
  4. This is 0 or 1, depending on the communication operation. For details, refer to **13.3 (12) Serial output register m (SOM)**.
  5. When using UART1 transmission and reception in a pair, set channel 2 of unit 0 to UART1 transmission (refer to **Table 13-7**).
  6. The SMR02 register of channel 2 of unit 0 must also be set during UART1 reception. For details, refer to **13.6.2 (1) Register setting**.
  7. Set the CKO03 bit to 1 before a start condition is generated. Clear the SO03 bit from 1 to 0 when the start condition is generated.
  8. Set the CKO03 bit to 1 before a stop condition is generated. Clear the SO03 bit from 0 to 1 when the stop condition is generated.

**Remark** X: Don't care



**Table 13-9. Relationship between register settings and pins  
(Channel 0 of unit 1: CSI20, UART2 transmission, IIC20)**

SE 10 Note1	MD 102	MD 101	SOE 10	SO 10	CKO 10	TXE 10	RXE 10	PM 142	P142	PM 143 Note2	P143 Note2	PM 144	P144	Operation mode	Pin Function			
															SCK20/ SCL20/P142	SI20/SDA20/ RxD2/P143 Note2	SO20/ TxD2/P144	
0	0	0	0	1	1	0	0	×	×	×	×	×	×	Operation stop mode	P142	P143	P144	
																P143/RxD2		
																P143		
1	0	0	0	1	1	0	1	1	×	1	×	×	×	Slave CSI20 reception	SCK20 (input)	SI20	P144	
			1	0/1 Note4	1	1	0	1	×	×	×	0	1	Slave CSI20 transmission	SCK20 (input)	P143	SO20	
			1	0/1 Note4	1	1	1	1	×	1	×	0	1	Slave CSI20 transmission/reception	SCK20 (input)	SI20	SO20	
			0	1	0/1 Note4	0	1	0	1	1	×	×	×	×	Master CSI20 reception	SCK20 (output)	SI20	P144
			1	0/1 Note4	0/1 Note4	1	0	0	1	×	×	×	0	1	Master CSI20 transmission	SCK20 (output)	P143	SO20
			1	0/1 Note4	0/1 Note4	1	1	0	1	1	×	×	0	1	Master CSI20 transmission/reception	SCK20 (output)	SI20	SO20
	0	1	1	0/1 Note4	1	1	0	×	×	×	×	0	1	UART2 transmission Note5	P142	P143/RxD2	TxD2	
0	1	0	0	0/1 Note6	0/1 Note6	0	0	0	1	0	1	×	×	IIC20 start condition	SCL20	SDA20	P144	
						1	0											
						0	1											
			1	1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	×	×	IIC20 address field transmission	SCL20	SDA20	P144
				1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	×	×	IIC20 data transmission	SCL20	SDA20	P144
0	1	0	0	0/1 Note7	0/1 Note7	0	0	0	1	0	1	×	×	IIC20 data reception	SCL20	SDA20	P144	
						1	0											
						0	1											

- Notes 1.** The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.
- When channel 1 of unit 1 is set to UART2 reception, this pin becomes an RxD2 function pin (refer to **Table 13-10**). In this case, operation stop mode or UART2 transmission must be selected for channel 0 of unit 1.
  - This pin can be set as a port function pin.
  - This is 0 or 1, depending on the communication operation. For details, refer to **13.3 (12) Serial output register m (SOM)**.
  - When using UART2 transmission and reception in a pair, set channel 1 of unit 1 to UART2 reception (refer to **Table 13-10**).
  - Set the CKO10 bit to 1 before a start condition is generated. Clear the SO10 bit from 1 to 0 when the start condition is generated.
  - Set the CKO10 bit to 1 before a stop condition is generated. Clear the SO10 bit from 0 to 1 when the stop condition is generated.

**Remark** X: Don't care

Table 13-10. Relationship between register settings and pins (Channel 1 of unit 1: CSI21, UART2 reception, IIC21)

SE 11 Note1	MD 112	MD 111	SOE 11	SO11	CKO 11	TXE 11	RXE 11	PM 125	P 125	PM 126	P 126	PM 127	P 127	PM 143 Note2	P 143 Note2	Operation mode	Pin Function																
																	SCK21/ SCL21/ P125	SI21/ SDA21/ P126	SO21/ P127	SI20/ SDA20/ RxD2/ P143 Note2													
0	0	0	0	1	1	0	0	× Note3	× Note3	× Note3	× Note3	× Note3	× Note3	× Note3	× Note3	Operation stop mode	P125	P126	P127	SI20/ SDA20/ P143 Note2													
	0	1																															
	1	0																															
1	0	0	0	1	1	0	1	1	×	1	×	× Note3	× Note3	× Note3	× Note3	Slave CSI21 reception	SCK21 (input)	SI21	P127	SI20/ SDA20/ P143 Note2													
			1	0/1 Note4	1	1	0	1	×	× Note3	× Note3	0	1	× Note3	× Note3	Slave CSI21 transmission	SCK21 (input)	P126	SO21	SI20/ SDA20/ P143 Note2													
			1	0/1 Note4	1	1	1	1	×	1	×	0	1	× Note3	× Note3	Slave CSI21 transmission /reception	SCK21 (input)	SI21	SO21	SI20/ SDA20/ P143 Note2													
			0	1	0/1 Note4	0	1	0	1	1	×	× Note3	× Note3	× Note3	× Note3	Master CSI21 reception	SCK21 (output)	SI21	P127	SI20/ SDA20/ P143 Note2													
			1	0/1 Note4	0/1 Note4	1	0	0	1	× Note3	× Note3	0	1	× Note3	× Note3	Master CSI21 transmission	SCK21 (output)	P126	SO21	SI20/ SDA20/ P143 Note2													
			1	0/1 Note4	0/1 Note4	1	1	0	1	1	×	0	1	× Note3	× Note3	Master CSI21 transmission /reception	SCK21 (output)	SI21	SO21	SI20/ SDA20/ P143 Note2													
			0	1	0	1	1	0	1	× Note3	× Note3	× Note3	× Note3	× Note3	× Note3	1	×	UART2 reception Notes5,6	P125	P126	P127	RxD2											
0	1	0	0	0/1 Note7	0/1 Note7	0	0	0	1	0	1	× Note3	× Note3	× Note3	× Note3	IIC21 start condition	SCL21	SDA21	P127	SI20/ SDA20/ P143 Note2													
						1	0																										
						0	1																										
1	1	1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	× Note3	× Note3	× Note3	× Note3	IIC21 address field transmission	SCL21	SDA21	P127	SI20/ SDA20/ P143 Note2														
																				1	0	0	1	0	1	× Note3	× Note3	× Note3	× Note3	IIC21 data transmission			
																				1	0/1 Note4	0/1 Note4	0	1	0	1	0	1	× Note3	× Note3	× Note3	× Note3	IIC21 data reception
																				0	0/1 Note8	0/1 Note8	0	0	0	1	0	1	× Note3	× Note3	× Note3	× Note3	IIC21 stop condition

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

- Notes**
1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.
  2. When channel 1 of unit 1 is set to UART2 reception, this pin becomes an RxD2 function pin. In this case, set channel 0 of unit 1 to operation stop mode or UART2 transmission (refer to **Table 13-9**).  
When channel 0 of unit 1 is set to CSI20 or IIC20, this pin cannot be used as an RxD2 function pin. In this case, set channel 2 of unit 1 to operation stop mode or CSI21 or IIC21.
  3. This pin can be set as a port function pin.
  4. This is 0 or 1, depending on the communication operation. For details, refer to **13.3 (12) Serial output register m (SOM)**.
  5. When using UART2 transmission and reception in a pair, set channel 1 of unit 0 to UART2 transmission (refer to **Table 13-9**).
  6. The SMR10 register of channel 1 of unit 0 must also be set during UART2 reception. For details, refer to **13.6.2 (1) Register setting**.
  7. Set the CKO11 bit to 1 before a start condition is generated. Clear the SO11 bit from 1 to 0 when the start condition is generated.
  8. Set the CKO11 bit to 1 before a stop condition is generated. Clear the SO11 bit from 0 to 1 when the stop condition is generated.

**Remark** X: Don't care

**Table 13-11. Relationship between register settings and pins (Channel 2 of unit 1: UART3 transmission)**

SE12 Note1	MD122	MD121	SOE12	SO12	TXE12	RXE12	PM13	P13	Operation mode	Pin Function
										EX27/TxD3/P13
0	0	1	0	1	0	0	×	×	Operation stop mode	EX27/P13
1	0	1	1	0/1 Note3	1	0	0	1	UART3 transmission Note4	TxD3

- Notes 1.** The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.
- This pin can be set as a port function pin.
  - This is 0 or 1, depending on the communication operation. For details, refer to **13.3 (12) Serial output register m (SOM)**.
  - When using UART3 transmission and reception in a pair, set channel 3 of unit 1 to UART3 reception (refer to **Table 13-12**).

**Remark** X: Don't care

**Table 13-12. Relationship between register settings and pins (Channel 3 of unit 1: UART3 reception)**

SE13 Note1	MD132	MD131	TXE13	RXE13	PM14	P14	Operation mode	Pin Function
								EX28/RxD3/P14
0	0	1	0	0	×	×	Operation stop mode	EX28/P14
1	0	1	0	1	1	×	UART3 reception Notes 3, 4	RxD3

- Notes 1.** The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.
- This pin can be set as a port function pin.
  - When using UART3 transmission and reception in a pair, set channel 2 of unit 1 to UART3 transmission (refer to **Table 13-11**).
  - The SMR12 register of channel 2 of unit 1 must also be set during UART3 reception. For details, refer to **13.6.2 (1) Register setting**.

**Remark** X: Don't care

## CHAPTER 14 SERIAL INTERFACE IIC0

### 14.1 Functions of Serial Interface IIC0

Serial interface IIC0 has the following two modes.

**(1) Operation stop mode**

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

**(2) I<sup>2</sup>C bus mode (multimaster supported)**

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL0) line and a serial data bus (SDA0) line.

This mode complies with the I<sup>2</sup>C bus format and the master device can generate “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the I<sup>2</sup>C bus.

Since the SCL0 and SDA0 pins are used for open drain outputs, IIC0 requires pull-up resistors for the serial clock line and the serial data bus line.

Figure 14-1 shows a block diagram of serial interface IIC0.

Figure 14-1. Block Diagram of Serial Interface IIC0

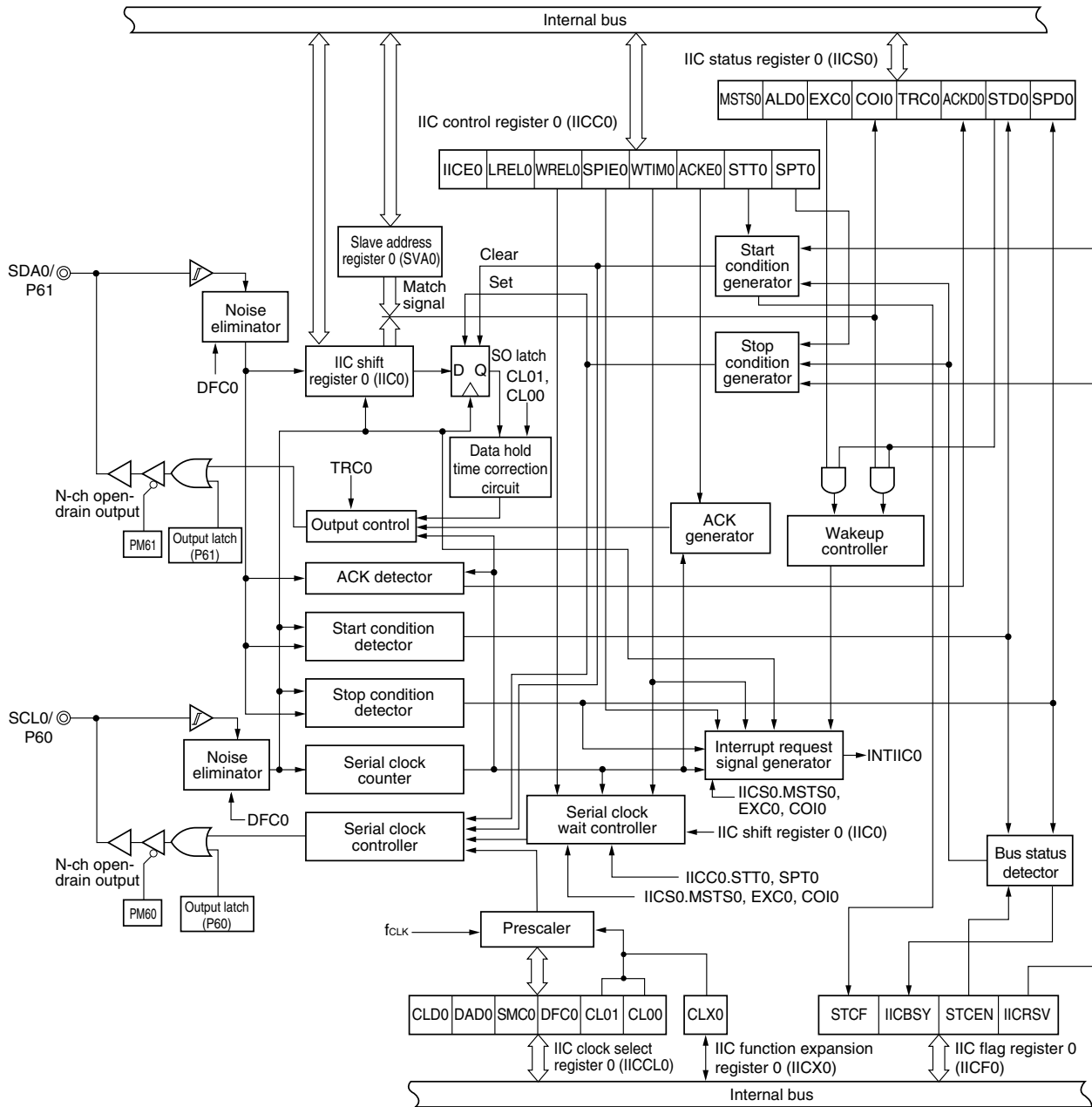
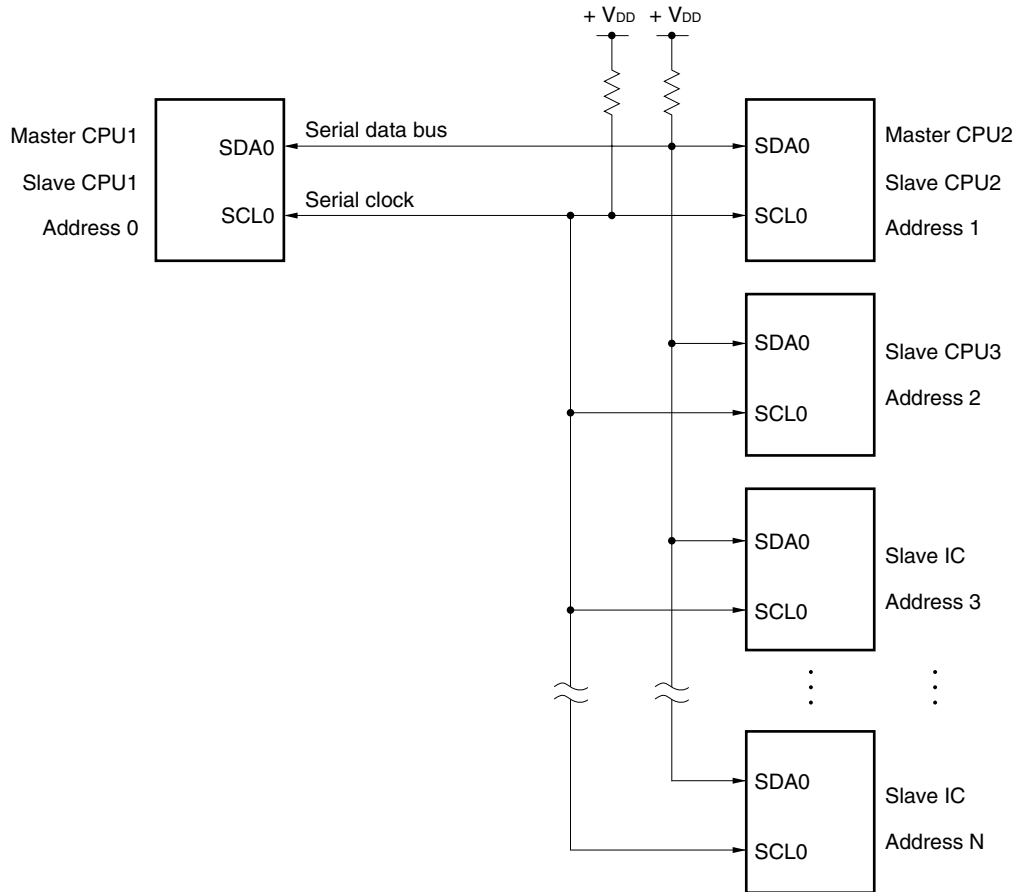


Figure 14-2 shows a serial bus configuration example.

**Figure 14-2. Serial Bus Configuration Example Using I<sup>2</sup>C Bus**



## 14.2 Configuration of Serial Interface IIC0

Serial interface IIC0 includes the following hardware.

**Table 14-1. Configuration of Serial Interface IIC0**

Item	Configuration
Registers	IIC shift register 0 (IIC0) Slave address register 0 (SVA0)
Control registers	Peripheral enable register 0 (PER0) IIC control register 0 (IICC0) IIC status register 0 (IICS0) IIC flag register 0 (IICF0) IIC clock select register 0 (IICCL0) IIC function expansion register 0 (IICX0) Port mode register 6 (PM6) Port register 6 (P6)

### (1) IIC shift register 0 (IIC0)

IIC0 is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. IIC0 can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to IIC0.

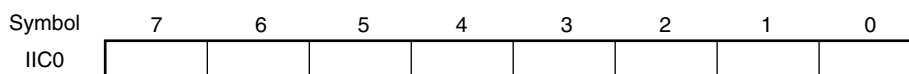
Cancel the wait state and start data transfer by writing data to IIC0 during the wait period.

IIC0 can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears IIC0 to 00H.

**Figure 14-3. Format of IIC Shift Register 0 (IIC0)**

Address: FFF50H After reset: 00H R/W



**Cautions 1. Do not write data to IIC0 during data transfer.**

- 2. Write or read IIC0 only during the wait period. Accessing IIC0 in a communication state other than during the wait period is prohibited. When the device serves as the master, however, IIC0 can be written only once after the communication trigger bit (STT0) is set to 1.**

### (2) Slave address register 0 (SVA0)

This register stores local addresses when in slave mode.

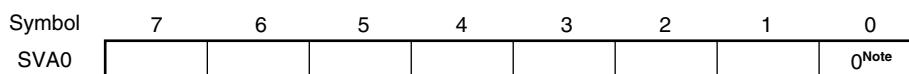
SVA0 can be set by an 8-bit memory manipulation instruction.

However, rewriting to this register is prohibited while STD0 = 1 (while the start condition is detected).

Reset signal generation clears SVA0 to 00H.

**Figure 14-4. Format of Slave Address Register 0 (SVA0)**

Address: FFF53H After reset: 00H R/W



**Note** Bit 0 is fixed to 0.



**(3) SO latch**

The SO latch is used to retain the SDA0 pin's output level.

**(4) Wakeup controller**

This circuit generates an interrupt request (INTIIC0) when the address received by this register matches the address value set to slave address register 0 (SVA0) or when an extension code is received.

**(5) Prescaler**

This selects the sampling clock to be used.

**(6) Serial clock counter**

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

**(7) Interrupt request signal generator**

This circuit controls the generation of interrupt request signals (INTIIC0).

An I<sup>2</sup>C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by SPIE0 bit)

**Remark** WTIM0 bit: Bit 3 of IIC control register 0 (IICC0)

SPIE0 bit: Bit 4 of IIC control register 0 (IICC0)

**(8) Serial clock controller**

In master mode, this circuit generates the clock output via the SCL0 pin from a sampling clock.

**(9) Serial clock wait controller**

This circuit controls the wait timing.

**(10) ACK generator, stop condition detector, start condition detector, and ACK detector**

These circuits generate and detect each status.

**(11) Data hold time correction circuit**

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

**(12) Start condition generator**

This circuit generates a start condition when the STT0 bit is set to 1.

However, in the communication reservation disabled status (IICRSV bit = 1), when the bus is not released (IICBSY bit = 1), start condition requests are ignored and the STCF bit is set to 1.

**(13) Stop condition generator**

This circuit generates a stop condition when the SPT0 bit is set to 1.

**(14) Bus status detector**

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions. However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCEN bit.

**Remark** STT0 bit: Bit 1 of IIC control register 0 (IICC0)  
SPT0 bit: Bit 0 of IIC control register 0 (IICC0)  
IICRSV bit: Bit 0 of IIC flag register 0 (IICF0)  
IICBSY bit: Bit 6 of IIC flag register 0 (IICF0)  
STCF bit: Bit 7 of IIC flag register 0 (IICF0)  
STCEN bit: Bit 1 of IIC flag register 0 (IICF0)

### 14.3 Registers to Controlling Serial Interface IIC0

Serial interface IIC0 is controlled by the following eight registers.

- Peripheral enable register 0 (PER0)
- IIC control register 0 (IICC0)
- IIC flag register 0 (IICF0)
- IIC status register 0 (IICS0)
- IIC clock select register 0 (IICCL0)
- IIC function expansion register 0 (IICX0)
- Port mode register 6 (PM6)
- Port register 6 (P6)

#### (1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial interface IIC0 is used, be sure to set bit 4 (IIC0EN) of this register to 1.

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

Reset signal generation clears this register to 00H.

**Figure 14-5. Format of Peripheral Enable Register 0 (PER0)**

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

IIC0EN	Control of serial interface IIC0 input clock
0	Stops supply of input clock. <ul style="list-style-type: none"> <li>• SFR used by serial interface IIC0 cannot be written.</li> <li>• Serial interface IIC0 is in the reset status.</li> </ul>
1	Supplies input clock. <ul style="list-style-type: none"> <li>• SFR used by serial interface IIC0 can be read/written.</li> </ul>

**Caution** When setting serial interface IIC0, be sure to set IIC0EN to 1 first. If IIC0EN = 0, writing to a control register of serial interface IIC0 is ignored, and, even if the register is read, only the default value is read (except for port mode register 6 (PM6) and port register 6 (P6)).

#### (2) IIC control register 0 (IICC0)

This register is used to enable/stop I<sup>2</sup>C operations, set wait timing, and set other I<sup>2</sup>C operations.

IICC0 can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE0, WTIM0, and ACKE0 bits while IICE0 bit = 0 or during the wait period. These bits can be set at the same time when the IICE0 bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Figure 14-6. Format of IIC Control Register 0 (IICC0) (1/4)

Address: FFF52H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC0	IICE0	LRELO	WRELO	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I <sup>2</sup> C operation enable
0	Stop operation. Reset IIC status register 0 (IICS0) <sup>Note 1</sup> . Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCL0 and SDA0 lines are at high level.	
Condition for clearing (IICE0 = 0)	
<ul style="list-style-type: none"> <li>• Cleared by instruction</li> <li>• Reset</li> </ul>	Condition for setting (IICE0 = 1)
<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>	

LRELO <sup>Note 2</sup>	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL0 and SDA0 lines are set to high impedance. The following flags of IIC control register 0 (IICC0) and IIC status register 0 (IICS0) are cleared to 0. • STT0 • SPT0 • MSTS0 • EXC0 • COI0 • TRC0 • ACKD0 • STD0
The standby mode following exit from communications remains in effect until the following communications entry conditions are met.	
<ul style="list-style-type: none"> <li>• After a stop condition is detected, restart is in master mode.</li> <li>• An address match or extension code reception occurs after the start condition.</li> </ul>	
Condition for clearing (LRELO = 0)	
<ul style="list-style-type: none"> <li>• Automatically cleared after execution</li> <li>• Reset</li> </ul>	Condition for setting (LRELO = 1)
<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>	

WRELO <sup>Note 2</sup>	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When WRELO is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC0 = 1), the SDA0 line goes into the high impedance state (TRC0 = 0).	
Condition for clearing (WRELO = 0)	
<ul style="list-style-type: none"> <li>• Automatically cleared after execution</li> <li>• Reset</li> </ul>	Condition for setting (WRELO = 1)
<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>	

- Notes 1.** The IICS0 register, the STCF and IICBSY bits of the IICF0 register, and the CLD0 and DAD0 bits of the IICCL0 register are reset.
- 2.** The signal of this bit is invalid while IICE0 is 0.

**Caution** The start condition is detected immediately after I<sup>2</sup>C is enabled to operate (IICE0 = 1) while the SCL0 line is at high level and the SDA0 line is at low level. Immediately after enabling I<sup>2</sup>C to operate (IICE0 = 1), set LRELO (1) by using a 1-bit memory manipulation instruction.

Figure 14-6. Format of IIC Control Register 0 (IICC0) (2/4)

SPIE0 <sup>Note 1</sup>	Enable/disable generation of interrupt request when stop condition is detected	
0	Disable	
1	Enable	
Condition for clearing (SPIE0 = 0)		Condition for setting (SPIE0 = 1)
<ul style="list-style-type: none"> <li>• Cleared by instruction</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>

WTIMO <sup>Note 1</sup>	Control of wait and interrupt request generation	
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.	
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and wait is set. Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device.	
An interrupt is generated at the falling edge of the ninth clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after an acknowledge ( $\overline{\text{ACK}}$ ) is issued. However, when the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.		
Condition for clearing (WTIMO = 0)		Condition for setting (WTIMO = 1)
<ul style="list-style-type: none"> <li>• Cleared by instruction</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>

ACKE0 <sup>Notes 1, 2</sup>	Acknowledgment control	
0	Disable acknowledgment.	
1	Enable acknowledgment. During the ninth clock period, the SDA0 line is set to low level.	
Condition for clearing (ACKE0 = 0)		Condition for setting (ACKE0 = 1)
<ul style="list-style-type: none"> <li>• Cleared by instruction</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>

- Notes**
1. The signal of this bit is invalid while IICE0 is 0. Set this bit during that period.
  2. The set value is invalid during address transfer and if the code is not an extension code. When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Figure 14-6. Format of IIC Control Register 0 (IICC0) (3/4)

<R>

STT0 <sup>Note</sup>	Start condition trigger	
0	Do not generate a start condition.	
1	<p>When bus is released (in standby state, when IICBSY = 0):                      Generate a start condition (for starting as master). When the SCL0 line is high level, the SDA0 line is changed from high level to low level and then the start condition is generated. Next, after the rated amount of time has elapsed, SCL0 is changed to low level (wait state).</p> <p>When a third party is communicating:</p> <ul style="list-style-type: none"> <li>• When communication reservation function is enabled (IICRSV = 0)                              Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released.</li> <li>• When communication reservation function is disabled (IICRSV = 1)                              STCF is set to 1 and STT0 is cleared. No start condition is generated.</li> </ul> <p>In the wait state (when master device):                      Generates a restart condition after releasing the wait.</p>	
<p>Cautions concerning set timing</p> <ul style="list-style-type: none"> <li>• For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when ACKE0 has been cleared to 0 and slave has been notified of final reception.</li> <li>• For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the wait period that follows output of the ninth clock.</li> <li>• Cannot be set to 1 at the same time as SPT0.</li> <li>• Setting STT0 to 1 and then setting it again before it is cleared to 0 is prohibited.</li> </ul>		
Condition for clearing (STT0 = 0)		Condition for setting (STT0 = 1)
<ul style="list-style-type: none"> <li>• Cleared by setting SST0 to 1 while communication reservation is prohibited.</li> <li>• Cleared by loss in arbitration</li> <li>• Cleared after start condition is generated by master device</li> <li>• Cleared by LREL0 = 1 (exit from communications)</li> <li>• When IICE0 = 0 (operation stop)</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>

**Note** The signal of this bit is invalid while IICE0 is 0.

- Remarks**
1. Bit 1 (STT0) becomes 0 when it is read after data setting.
  2. IICRSV: Bit 0 of IIC flag register (IICF0)  
 STCF: Bit 7 of IIC flag register (IICF0)

Figure 14-6. Format of IIC Control Register 0 (IICC0) (4/4)

SPT0	Stop condition trigger				
0	Stop condition is not generated.				
1	Stop condition is generated (termination of master device's transfer). After the SDA0 line goes to low level, either set the SCL0 line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDA0 line changes from low level to high level and a stop condition is generated.				
Cautions concerning set timing <ul style="list-style-type: none"> <li>• For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when ACKE0 has been cleared to 0 and slave has been notified of final reception.</li> <li>• For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the wait period that follows output of the ninth clock.</li> <li>• Cannot be set to 1 at the same time as STT0.</li> <li>• SPT0 can be set to 1 only when in master mode<sup>Note</sup>.</li> <li>• When WTIM0 has been cleared to 0, if SPT0 is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. WTIM0 should be changed from 0 to 1 during the wait period following the output of eight clocks, and SPT0 should be set to 1 during the wait period that follows the output of the ninth clock.</li> <li>• Setting SPT0 to 1 and then setting it again before it is cleared to 0 is prohibited.</li> </ul>					
<table border="1"> <thead> <tr> <th>Condition for clearing (SPT0 = 0)</th> <th>Condition for setting (SPT0 = 1)</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> <li>• Cleared by loss in arbitration</li> <li>• Automatically cleared after stop condition is detected</li> <li>• Cleared by LREL0 = 1 (exit from communications)</li> <li>• When IICE0 = 0 (operation stop)</li> <li>• Reset</li> </ul> </td> <td> <ul style="list-style-type: none"> <li>• Set by instruction</li> </ul> </td> </tr> </tbody> </table>		Condition for clearing (SPT0 = 0)	Condition for setting (SPT0 = 1)	<ul style="list-style-type: none"> <li>• Cleared by loss in arbitration</li> <li>• Automatically cleared after stop condition is detected</li> <li>• Cleared by LREL0 = 1 (exit from communications)</li> <li>• When IICE0 = 0 (operation stop)</li> <li>• Reset</li> </ul>	<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>
Condition for clearing (SPT0 = 0)	Condition for setting (SPT0 = 1)				
<ul style="list-style-type: none"> <li>• Cleared by loss in arbitration</li> <li>• Automatically cleared after stop condition is detected</li> <li>• Cleared by LREL0 = 1 (exit from communications)</li> <li>• When IICE0 = 0 (operation stop)</li> <li>• Reset</li> </ul>	<ul style="list-style-type: none"> <li>• Set by instruction</li> </ul>				

**Note** Set SPT0 to 1 only in master mode. However, SPT0 must be set to 1 and a stop condition generated before the first stop condition is detected following the switch to the operation enabled status.

**Caution** When bit 3 (TRC0) of IIC status register 0 (IICS0) is set to 1, WREL0 is set to 1 during the ninth clock and wait is canceled, after which TRC0 is cleared and the SDA0 line is set to high impedance.

**Remark** Bit 0 (SPT0) becomes 0 when it is read after data setting.

**(3) IIC status register 0 (IICS0)**

This register indicates the status of I<sup>2</sup>C.

IICS0 is read by a 1-bit or 8-bit memory manipulation instruction only when STT0 = 1 and during the wait period.

Reset signal generation clears this register to 00H.

**Figure 14-7. Format of IIC Status Register 0 (IICS0) (1/3)**

Address: FFF56H      After reset: 00H      R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master device status	
0	Slave device status or communication standby status	
1	Master device communication status	
Condition for clearing (MSTS0 = 0)		Condition for setting (MSTS0 = 1)
<ul style="list-style-type: none"> <li>• When a stop condition is detected</li> <li>• When ALD0 = 1 (arbitration loss)</li> <li>• Cleared by LREL0 = 1 (exit from communications)</li> <li>• When IICE0 changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• When a start condition is generated</li> </ul>

ALD0	Detection of arbitration loss	
0	This status means either that there was no arbitration or that the arbitration result was a "win".	
1	This status indicates the arbitration result was a "loss". MSTS0 is cleared.	
Condition for clearing (ALD0 = 0)		Condition for setting (ALD0 = 1)
<ul style="list-style-type: none"> <li>• Automatically cleared after IICS0 is read<sup>Note</sup></li> <li>• When IICE0 changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• When the arbitration result is a "loss".</li> </ul>

EXC0	Detection of extension code reception	
0	Extension code was not received.	
1	Extension code was received.	
Condition for clearing (EXC0 = 0)		Condition for setting (EXC0 = 1)
<ul style="list-style-type: none"> <li>• When a start condition is detected</li> <li>• When a stop condition is detected</li> <li>• Cleared by LREL0 = 1 (exit from communications)</li> <li>• When IICE0 changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• When the higher four bits of the received address data is either "0000" or "1111" (set at the rising edge of the eighth clock).</li> </ul>

**Note** This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than IICS0. Therefore, when using the ALD0 bit, read the data of this bit before the data of the other bits.

**Remark** LREL0: Bit 6 of IIC control register 0 (IICC0)  
 IICE0: Bit 7 of IIC control register 0 (IICC0)



Figure 14-7. Format of IIC Status Register 0 (IICS0) (2/3)

COI0	Detection of matching addresses	
0	Addresses do not match.	
1	Addresses match.	
Condition for clearing (COI0 = 0)		Condition for setting (COI0 = 1)
<ul style="list-style-type: none"> <li>• When a start condition is detected</li> <li>• When a stop condition is detected</li> <li>• Cleared by LREL0 = 1 (exit from communications)</li> <li>• When IICE0 changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• When the received address matches the local address (slave address register 0 (SVA0)) (set at the rising edge of the eighth clock).</li> </ul>

TRC0	Detection of transmit/receive status	
0	Receive status (other than transmit status). The SDA0 line is set for high impedance.	
1	Transmit status. The value in the SO0 latch is enabled for output to the SDA0 line (valid starting at the falling edge of the first byte's ninth clock).	
Condition for clearing (TRC0 = 0)		Condition for setting (TRC0 = 1)
<p>&lt;Both master and slave&gt;</p> <ul style="list-style-type: none"> <li>• When a stop condition is detected</li> <li>• Cleared by LREL0 = 1 (exit from communications)</li> <li>• When IICE0 changes from 1 to 0 (operation stop)</li> <li>• Cleared by WREL0 = 1<sup>Note</sup> (wait cancel)</li> <li>• When ALD0 changes from 0 to 1 (arbitration loss)</li> <li>• Reset</li> </ul> <p>&lt;Master&gt;</p> <ul style="list-style-type: none"> <li>• When "1" is output to the first byte's LSB (transfer direction specification bit)</li> </ul> <p>&lt;Slave&gt;</p> <ul style="list-style-type: none"> <li>• When a start condition is detected</li> <li>• When "0" is input to the first byte's LSB (transfer direction specification bit)</li> </ul> <p>&lt;When not used for communication&gt;</p>		<p>&lt;Master&gt;</p> <ul style="list-style-type: none"> <li>• When a start condition is generated</li> <li>• When "0" is output to the first byte's LSB (transfer direction specification bit)</li> </ul> <p>&lt;Slave&gt;</p> <ul style="list-style-type: none"> <li>• When "1" is input to the first byte's LSB (transfer direction specification bit)</li> </ul>

**Note** If the wait state is canceled by setting bit 5 (WREL0) of IIC control register 0 (IICC0) to 1 at the ninth clock when bit 3 (TRC0) of IIC status register 0 (IICS0) is 1, TRC0 is cleared, and the SDA0 line goes into a high-impedance state.

**Remark** LREL0: Bit 6 of IIC control register 0 (IICC0)  
IICE0: Bit 7 of IIC control register 0 (IICC0)

Figure 14-7. Format of IIC Status Register 0 (IICS0) (3/3)

ACKD0	Detection of acknowledge ( $\overline{\text{ACK}}$ )	
0	Acknowledge was not detected.	
1	Acknowledge was detected.	
Condition for clearing (ACKD0 = 0)		Condition for setting (ACKD0 = 1)
<ul style="list-style-type: none"> <li>• When a stop condition is detected</li> <li>• At the rising edge of the next byte's first clock</li> <li>• Cleared by LREL0 = 1 (exit from communications)</li> <li>• When IICE0 changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• After the SDA0 line is set to low level at the rising edge of SCL0's ninth clock</li> </ul>

STD0	Detection of start condition	
0	Start condition was not detected.	
1	Start condition was detected. This indicates that the address transfer period is in effect.	
Condition for clearing (STD0 = 0)		Condition for setting (STD0 = 1)
<ul style="list-style-type: none"> <li>• When a stop condition is detected</li> <li>• At the rising edge of the next byte's first clock following address transfer</li> <li>• Cleared by LREL0 = 1 (exit from communications)</li> <li>• When IICE0 changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• When a start condition is detected</li> </ul>

SPD0	Detection of stop condition	
0	Stop condition was not detected.	
1	Stop condition was detected. The master device's communication is terminated and the bus is released.	
Condition for clearing (SPD0 = 0)		Condition for setting (SPD0 = 1)
<ul style="list-style-type: none"> <li>• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition</li> <li>• When IICE0 changes from 1 to 0 (operation stop)</li> <li>• Reset</li> </ul>		<ul style="list-style-type: none"> <li>• When a stop condition is detected</li> </ul>

**Remark** LREL0: Bit 6 of IIC control register 0 (IICC0)

IICE0: Bit 7 of IIC control register 0 (IICC0)

#### (4) IIC flag register 0 (IICF0)

This register sets the operation mode of I<sup>2</sup>C and indicates the status of the I<sup>2</sup>C bus.

IICF0 can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STCF and IICBSY bits are read-only.

The IICRSV bit can be used to enable/disable the communication reservation function.

STCEN can be used to set the initial value of the IICBSY bit.

IICRSV and STCEN can be written only when the operation of I<sup>2</sup>C is disabled (bit 7 (IICE0) of IIC control register 0 (IICC0) = 0). When operation is enabled, the IICF0 register can be read.

Reset signal generation clears this register to 00H.

Figure 14-8. Format of IIC Flag Register 0 (IICF0)

Address: FFF51H    After reset: 00H    R/W<sup>Note</sup>

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	STT0 clear flag	
0	Generate start condition	
1	Start condition generation unsuccessful: clear STT0 flag	
Condition for clearing (STCF = 0)		Condition for setting (STCF = 1)
<ul style="list-style-type: none"> <li>Cleared by STT0 = 1</li> <li>When IICE0 = 0 (operation stop)</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>Generating start condition unsuccessful and STT0 cleared to 0 when communication reservation is disabled (IICRSV = 1).</li> </ul>

IICBSY	I <sup>2</sup> C bus status flag	
0	Bus release status (communication initial status when STCEN = 1)	
1	Bus communication status (communication initial status when STCEN = 0)	
Condition for clearing (IICBSY = 0)		Condition for setting (IICBSY = 1)
<ul style="list-style-type: none"> <li>Detection of stop condition</li> <li>When IICE0 = 0 (operation stop)</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>Detection of start condition</li> <li>Setting of IICE0 when STCEN = 0</li> </ul>

STCEN	Initial start enable trigger	
0	After operation is enabled (IICE0 = 1), enable generation of a start condition upon detection of a stop condition.	
1	After operation is enabled (IICE0 = 1), enable generation of a start condition without detecting a stop condition.	
Condition for clearing (STCEN = 0)		Condition for setting (STCEN = 1)
<ul style="list-style-type: none"> <li>Cleared by instruction</li> <li>Detection of start condition</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>Set by instruction</li> </ul>

IICRSV	Communication reservation function disable bit	
0	Enable communication reservation	
1	Disable communication reservation	
Condition for clearing (IICRSV = 0)		Condition for setting (IICRSV = 1)
<ul style="list-style-type: none"> <li>Cleared by instruction</li> <li>Reset</li> </ul>		<ul style="list-style-type: none"> <li>Set by instruction</li> </ul>

**Note** Bits 6 and 7 are read-only.

- Cautions**
1. Write to STCEN only when the operation is stopped (IICE0 = 0).
  2. As the bus release status (IICBSY = 0) is recognized regardless of the actual bus status when STCEN = 1, when generating the first start condition (STT0 = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.
  3. Write to IICRSV only when the operation is stopped (IICE0 = 0).

**Remark** STT0: Bit 1 of IIC control register 0 (IICC0)  
 IICE0: Bit 7 of IIC control register 0 (IICC0)

**(5) IIC clock select register 0 (IICCL0)**

This register is used to set the transfer clock for the I<sup>2</sup>C bus.

IICCL0 can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD0 and DAD0 bits are read-only. The SMC0, CL01, and CL00 bits are set in combination with bit 0 (CLX0) of IIC function expansion register 0 (IICX0) (see **14.5.4 Transfer clock setting method**).

Set IICCL0 while bit 7 (IICE0) of IIC control register 0 (IICC0) is 0.

Reset signal generation clears this register to 00H.

**Figure 14-9. Format of IIC Clock Select Register 0 (IICCL0)**

Address: FFF54H    After reset: 00H    R/W<sup>Note</sup>

Symbol	7	6	<5>	<4>	<3>	<2>	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	CL01	CL00

CLD0	Detection of SCL0 pin level (valid only when IICE0 = 1)
0	The SCL0 pin was detected at low level.
1	The SCL0 pin was detected at high level.
Condition for clearing (CLD0 = 0)	
<ul style="list-style-type: none"> <li>• When the SCL0 pin is at low level</li> <li>• When IICE0 = 0 (operation stop)</li> <li>• Reset</li> </ul>	
Condition for setting (CLD0 = 1)	
<ul style="list-style-type: none"> <li>• When the SCL0 pin is at high level</li> </ul>	

DAD0	Detection of SDA0 pin level (valid only when IICE0 = 1)
0	The SDA0 pin was detected at low level.
1	The SDA0 pin was detected at high level.
Condition for clearing (DAD0 = 0)	
<ul style="list-style-type: none"> <li>• When the SDA0 pin is at low level</li> <li>• When IICE0 = 0 (operation stop)</li> <li>• Reset</li> </ul>	
Condition for setting (DAD0 = 1)	
<ul style="list-style-type: none"> <li>• When the SDA0 pin is at high level</li> </ul>	

SMC0	Operation mode switching
0	Operates in standard mode.
1	Operates in fast mode.

DFC0	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in fast mode. In fast mode, the transfer clock does not vary regardless of DFC0 bit set (1)/clear (0). The digital filter is used for noise elimination in fast mode.	

**Note** Bits 4 and 5 are read-only.

**Remark** IICE0: Bit 7 of IIC control register 0 (IICC0)

**(6) IIC function expansion register 0 (IICX0)**

This register sets the function expansion of I<sup>2</sup>C.

IICX0 can be set by a 1-bit or 8-bit memory manipulation instruction. The CLX0 bit is set in combination with bits 3, 1, and 0 (SMC0, CL01, and CL00) of IIC clock select register 0 (IICCL0) (see **14.5.4 Transfer clock setting method**).

Set IICX0 while bit 7 (IICE0) of IIC control register 0 (IICC0) is 0.

Reset signal generation clears this register to 00H.

**Figure 14-10. Format of IIC Function Expansion Register 0 (IICX0)**

Address: FFF55H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	<0>
IICX0	0	0	0	0	0	0	0	CLX0

**Table 14-2. Selection Clock Setting**

IICX0	IICCL0			Transfer Clock (f <sub>CLK</sub> /m)	Settable Selection Clock (f <sub>CLK</sub> ) Range	Operation Mode
	Bit 3	Bit 1	Bit 0			
CLX0	SMC0	CL01	CL00			
0	0	0	0	f <sub>CLK</sub> /88	4.00 MHz to 8.4 MHz	Normal mode (SMC0 bit = 0)
0	0	0	1	f <sub>CLK</sub> /172	8.38 MHz to 16.76 MHz	
0	0	1	0	f <sub>CLK</sub> /344	16.76 MHz to 20 MHz	
0	0	1	1	f <sub>CLK</sub> /44	2.00 MHz to 4.2 MHz	
0	1	0	×	f <sub>CLK</sub> /48	7.60 MHz to 16.76 MHz	Fast mode (SMC0 bit = 1)
0	1	1	0	f <sub>CLK</sub> /96	16.00 MHz to 20 MHz	
0	1	1	1	f <sub>CLK</sub> /24	4.00 MHz to 8.4 MHz	
1	0	×	×	Setting prohibited		
1	1	0	×	f <sub>CLK</sub> /48	8.00 MHz to 8.38 MHz	Fast mode (SMC0 bit = 1)
1	1	1	0	Setting prohibited		
1	1	1	1	f <sub>CLK</sub> /24	4.00 MHz to 4.19 MHz	

**Caution** Determine the transfer clock frequency of I<sup>2</sup>C by using CLX0, SMC0, CL01, and CL00 before enabling the operation (by setting bit 7 (IICE0) of IIC control register 0 (IICC0) to 1). To change the transfer clock frequency, clear IICE0 once to 0.

- Remarks**
1. ×: don't care
  2. f<sub>CLK</sub>: CPU/peripheral hardware clock frequency

**(7) Port mode register 6 (PM6)**

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCL0 pin as clock I/O and the P61/SDA0 pin as serial data I/O, clear PM60 and PM61, and the output latches of P60 and P61 to 0.

Set IICE0 (bit 7 of IIC control register 0 (IICC0)) to 1 before setting the output mode because the P60/SCL0 and P61/SDA0 pins output a low level (fixed) when IICE0 is 0.

PM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

**Figure 14-11. Format of Port Mode Register 6 (PM6)**

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

## 14.4 I<sup>2</sup>C Bus Mode Functions

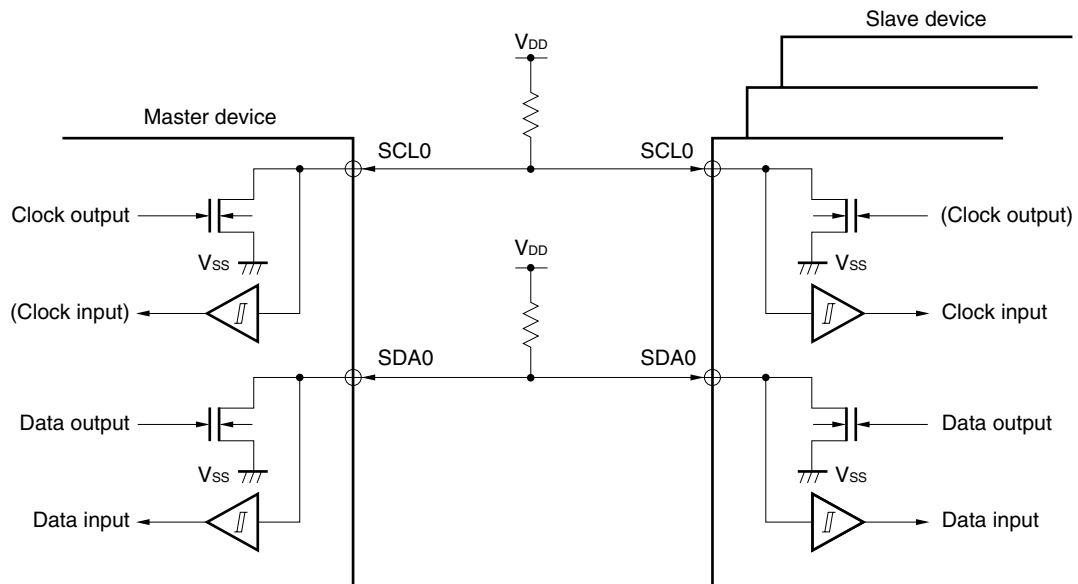
### 14.4.1 Pin configuration

The serial clock pin (SCL0) and serial data bus pin (SDA0) are configured as follows.

- (1) SCL0 ..... This pin is used for serial clock input and output.  
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.
- (2) SDA0 ..... This pin is used for serial data input and output.  
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

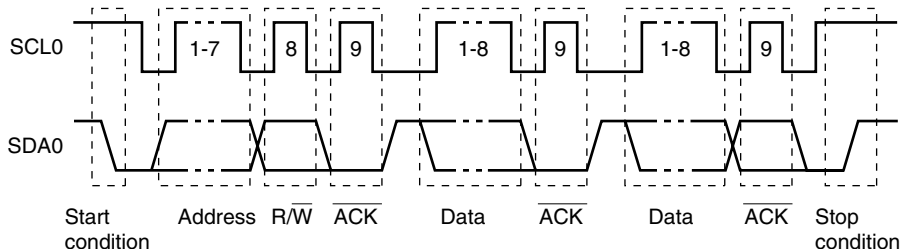
**Figure 14-12. Pin Configuration Diagram**



## 14.5 I<sup>2</sup>C Bus Definitions and Control Methods

The following section describes the I<sup>2</sup>C bus's serial data communication format and the signals used by the I<sup>2</sup>C bus. Figure 14-13 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I<sup>2</sup>C bus's serial data bus.

**Figure 14-13. I<sup>2</sup>C Bus Serial Data Transfer Timing**



The master device generates the start condition, slave address, and stop condition.

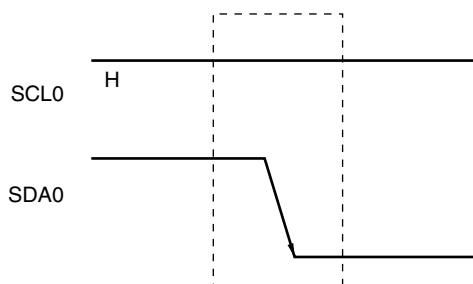
The acknowledge ( $\overline{\text{ACK}}$ ) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCL0) is continuously output by the master device. However, in the slave device, the SCL0's low level period can be extended and a wait can be inserted.

### 14.5.1 Start conditions

A start condition is met when the SCL0 pin is at high level and the SDA0 pin changes from high level to low level. The start conditions for the SCL0 pin and SDA0 pin are signals that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

**Figure 14-14. Start Conditions**



A start condition is output when bit 1 (STT0) of IIC control register 0 (IICC0) is set (to 1) after a stop condition has been detected (SPD0: Bit 0 = 1 in IIC status register 0 (IICS0)). When a start condition is detected, bit 1 (STD0) of IICS0 is set (to 1).



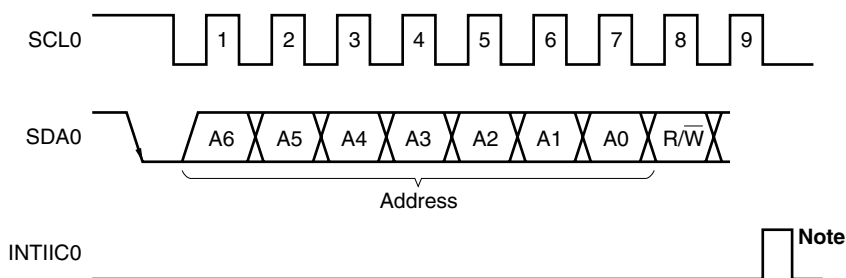
### 14.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in slave address register 0 (SVA0). If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 14-15. Address



**Note** INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

The slave address and the eighth bit, which specifies the transfer direction as described in **14.5.3 Transfer direction specification** below, are together written to IIC shift register 0 (IIC0) and are then output. Received addresses are written to IIC0.

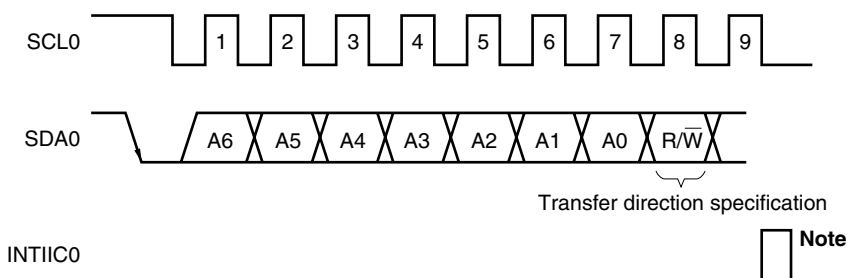
The slave address is assigned to the higher 7 bits of IIC0.

### 14.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit has a value of “0”, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of “1”, it indicates that the master device is receiving data from a slave device.

Figure 14-16. Transfer Direction Specification



**Note** INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

#### 14.5.4 Transfer clock setting method

##### (1) Selection clock setting method on the master side

The I<sup>2</sup>C transfer clock frequency ( $f_{SCL}$ ) is calculated using the following expression.

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

$m = 24, 44, 48, 88, 96, 172, 344$  (see **Table 14-3 Selection Clock Setting**)

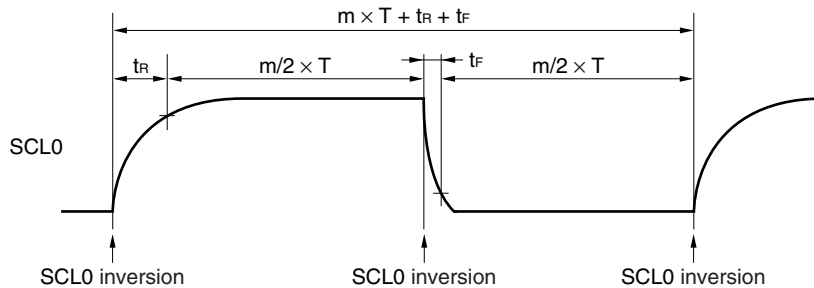
$T$ :  $1/f_{CLK}$

$t_R$ : SCL0 rise time

$t_F$ : SCL0 fall time

For example, the I<sup>2</sup>C transfer clock frequency ( $f_{SCL}$ ) when  $f_{CLK} = 4.19$  MHz,  $m = 88$ ,  $t_R = 200$  ns, and  $t_F = 50$  ns is calculated using following expression.

$$f_{SCL} = 1/(88 \times 238.7 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 47.0 \text{ kHz}$$



The selection clock is set using a combination of bits 3, 1, and 0 (SMC0, CL01, and CL00) of IIC clock select register 0 (IICCL0) and bit 0 (CLX0) of IIC function expansion register 0 (IICX0).

##### (2) Selection clock setting method on the slave side

To use as slave, set the bits 3, 1, and 0 (SMC0, CL01, CL00) of the IIC clock selection register (IICL0) and the bit 0 (CLX0) of the IIC function expansion register 0 (IICX0) according to the  $f_{CLK}$  (Selectable Selection Clock Range) and IIC Operation Mode (Normal or Fast) as defined in **Table 14-3. Selection Clock Setting**.

Table 14-3. Selection Clock Setting

IICX0	IICCL0			Transfer Clock ( $f_{CLK/m}$ )	Settable Selection Clock ( $f_{CLK}$ ) Range	Operation Mode
	Bit 0	Bit 3	Bit 1			
CLX0	SMC0	CL01	CL00			
0	0	0	0	$f_{CLK}/88$	4.00 MHz to 8.4 MHz	Normal mode (SMC0 bit = 0)
0	0	0	1	$f_{CLK}/172$	8.38 MHz to 16.76 MHz	
0	0	1	0	$f_{CLK}/344$	16.76 MHz to 20 MHz	
0	0	1	1	$f_{CLK}/44$	2.00 MHz to 4.2 MHz	
0	1	0	×	$f_{CLK}/48$	7.60 MHz to 16.76 MHz	Fast mode (SMC0 bit = 1)
0	1	1	0	$f_{CLK}/96$	16.00 MHz to 20 MHz	
0	1	1	1	$f_{CLK}/24$	4.00 MHz to 8.4 MHz	
1	0	×	×	Setting prohibited		
1	1	0	×	$f_{CLK}/48$	8.00 MHz to 8.38 MHz	Fast mode (SMC0 bit = 1)
1	1	1	0	Setting prohibited	16.00 MHz to 16.76 MHz	
1	1	1	1	$f_{CLK}/24$	4.00 MHz to 4.19 MHz	

**Caution** Determine the transfer clock frequency of I<sup>2</sup>C by using CLX0, SMC0, CL01, and CL00 before enabling the operation (by setting bit 7 (IICE0) of IIC control register 0 (IICC0) to 1). To change the transfer clock frequency, clear IICE0 once to 0.

- Remarks**
1. ×: don't care
  2.  $f_{CLK}$ : CPU/peripheral hardware clock frequency

#### 14.5.5 Acknowledge ( $\overline{ACK}$ )

$\overline{ACK}$  is used to check the status of serial data at the transmission and reception sides.

The reception side returns  $\overline{ACK}$  each time it has received 8-bit data.

The transmission side usually receives  $\overline{ACK}$  after transmitting 8-bit data. When  $\overline{ACK}$  is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether  $\overline{ACK}$  has been detected can be checked by using bit 2 (ACKD0) of IIC status register 0 (IICS0).

When the master receives the last data item, it does not return  $\overline{ACK}$  and instead generates a stop condition. If a slave does not return  $\overline{ACK}$  after receiving data, the master outputs a stop condition or restart condition and stops transmission. If  $\overline{ACK}$  is not returned, the possible causes are as follows.

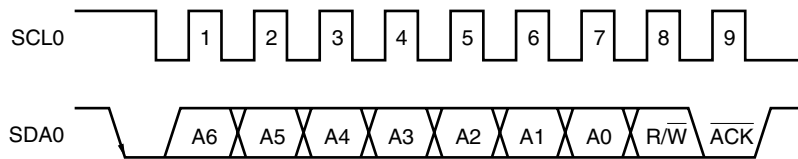
- <1> Reception was not performed normally.
- <2> The final data item was received.
- <3> The reception side specified by the address does not exist.

To generate  $\overline{ACK}$ , the reception side makes the SDA0 line low at the ninth clock (indicating normal reception).

Automatic generation of  $\overline{ACK}$  is enabled by setting bit 2 (ACKE0) of IIC control register 0 (IICC0) to 1. Bit 3 (TRC0) of the IICS0 register is set by the data of the eighth bit that follows 7-bit address information. Usually, set ACEK0 to 1 for reception (TRC0 = 0).

If a slave can receive no more data during reception (TRC0 = 0) or does not require the next data item, then the slave must inform the master, by clearing ACEK0 to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC0 = 0), it must clear ACEK0 to 0 so that  $\overline{ACK}$  is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 14-17.  $\overline{ACK}$ 

When the local address is received,  $\overline{ACK}$  is automatically generated, regardless of the value of ACKE0. When an address other than that of the local address is received,  $\overline{ACK}$  is not generated (NACK).

When an extension code is received,  $\overline{ACK}$  is generated if ACKE0 is set to 1 in advance.

How  $\overline{ACK}$  is generated when data is received differs as follows depending on the setting of the wait timing.

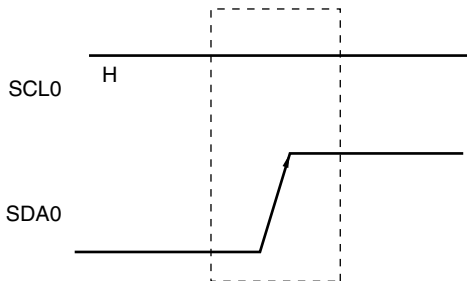
- When 8-clock wait state is selected (bit 3 (WTIM0) of IICC0 register = 0):  
By setting ACKE0 to 1 before releasing the wait state,  $\overline{ACK}$  is generated at the falling edge of the eighth clock of the SCL0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM0) of IICC0 register = 1):  
 $\overline{ACK}$  is generated by setting ACKE0 to 1 in advance.

### 14.5.6 Stop condition

When the SCL0 pin is at high level, changing the SDA0 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

**Figure 14-18. Stop Condition**



A stop condition is generated when bit 0 (SPT0) of IIC control register 0 (IICC0) is set to 1. When the stop condition is detected, bit 0 (SPD0) of IIC status register 0 (IICS0) is set to 1 and INTIIC0 is generated when bit 4 (SPIE0) of IICC0 is set to 1.

14.5.7 Wait

The wait is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL0 pin to low level notifies the communication partner of the wait state. When wait state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 14-19. Wait (1/2)

(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKE0 = 1)

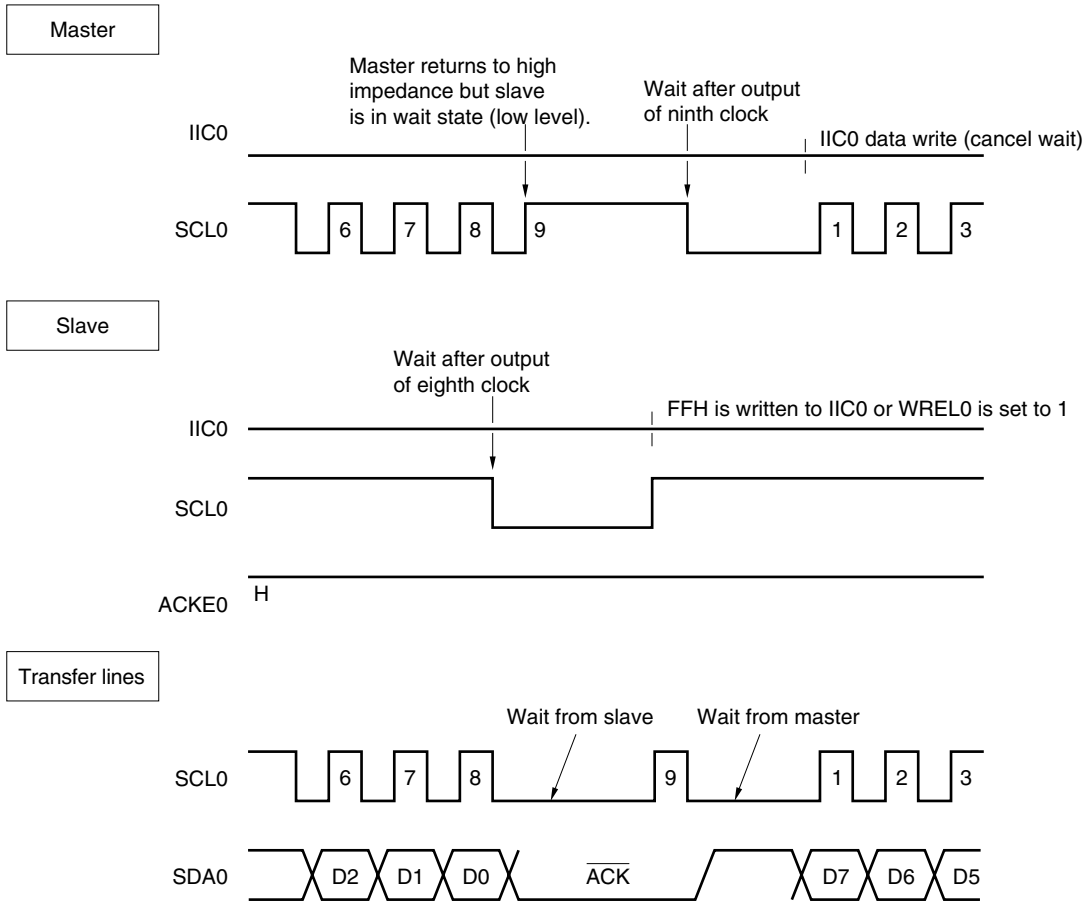
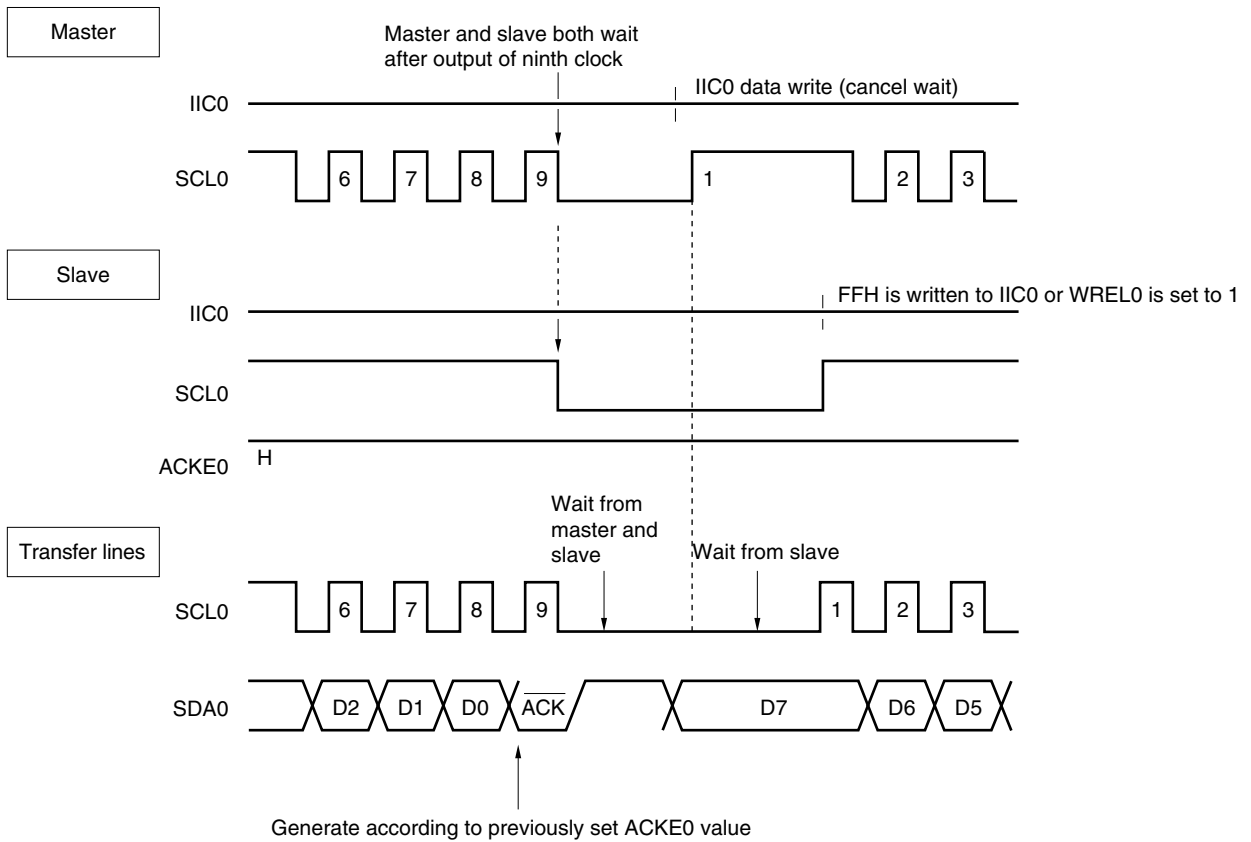


Figure 14-19. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE0 = 1)



**Remark** ACKE0: Bit 2 of IIC control register 0 (IICC0)  
WRELO: Bit 5 of IIC control register 0 (IICC0)

A wait may be automatically generated depending on the setting of bit 3 (WTIM0) of IIC control register 0 (IICC0). Normally, the receiving side cancels the wait state when bit 5 (WRELO) of IICC0 is set to 1 or when FFH is written to IIC shift register 0 (IIC0), and the transmitting side cancels the wait state when data is written to IIC0.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT0) of IICC0 to 1
- By setting bit 0 (SPT0) of IICC0 to 1

### 14.5.8 Canceling wait

The I<sup>2</sup>C usually cancels a wait state by the following processing.

- Writing data to IIC shift register 0 (IIC0)
- Setting bit 5 (WREL0) of IIC control register 0 (IICC0) (canceling wait)
- Setting bit 1 (STT0) of IIC0 register (generating start condition)<sup>Note</sup>
- Setting bit 0 (SPT0) of IIC0 register (generating stop condition)<sup>Note</sup>

**Note** Master only

When the above wait canceling processing is executed, the I<sup>2</sup>C cancels the wait state and communication is resumed.

To cancel a wait state and transmit data (including addresses), write the data to IIC0.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL0) of the IIC0 control register 0 (IICC0) to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT0) of IICC0 to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT0) of IICC0 to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to IIC0 after canceling a wait state by setting WREL0 to 1, an incorrect value may be output to SDA0 because the timing for changing the SDA0 line conflicts with the timing for writing IIC0.

In addition to the above, communication is stopped if IICE0 is cleared to 0 when communication has been aborted, so that the wait state can be canceled.

If the I<sup>2</sup>C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LREL0) of IICC0, so that the wait state can be canceled.



### 14.5.9 Interrupt request (INTIIC0) generation timing and wait control

The setting of bit 3 (WTIM0) of IIC control register 0 (IICC0) determines the timing by which INTIIC0 is generated and the corresponding wait control, as shown in Table 14-4.

**Table 14-4. INTIIC0 Generation Timing and Wait Control**

WTIM0	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	g <sup>Notes 1, 2</sup>	g <sup>Note 2</sup>	g <sup>Note 2</sup>	9	8	8
1	g <sup>Notes 1, 2</sup>	g <sup>Note 2</sup>	g <sup>Note 2</sup>	9	9	9

**Notes 1.** The slave device's INTIIC0 signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to slave address register 0 (SVA0).

At this point,  $\overline{ACK}$  is generated regardless of the value set to IICC0's bit 2 (ACKE0). For a slave device that has received an extension code, INTIIC0 occurs at the falling edge of the eighth clock.

However, if the address does not match after restart, INTIIC0 is generated at the falling edge of the 9th clock, but wait does not occur.

- 2.** If the received address does not match the contents of slave address register 0 (SVA0) and extension code is not received, neither INTIIC0 nor a wait occurs.

**Remark** The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

#### (1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIM0 bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

#### (2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

#### (3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

#### (4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to IIC shift register 0 (IIC0)
- Setting bit 5 (WRELO) of IIC control register 0 (IICC0) (canceling wait)
- Setting bit 1 (STT0) of IIC0 register (generating start condition)<sup>Note</sup>
- Setting bit 0 (SPT0) of IIC0 register (generating stop condition)<sup>Note</sup>

**Note** Master only.

When an 8-clock wait has been selected (WTIM0 = 0), the presence/absence of  $\overline{ACK}$  generation must be determined prior to wait cancellation.

#### (5) Stop condition detection

INTIIC0 is generated when a stop condition is detected (only when SPIE0 = 1).

#### 14.5.10 Address match detection method

In I<sup>2</sup>C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An INTIIC0 occurs when the address set to the slave address register 0 (SVA0) matches the slave address sent by the master device, or when an extension code has been received.

#### 14.5.11 Error detection

In I<sup>2</sup>C bus mode, the status of the serial data bus (SDA0) during data transmission is captured by IIC shift register 0 (IIC0) of the transmitting device, so the IIC0 data prior to transmission can be compared with the transmitted IIC0 data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

#### 14.5.12 Extension code

- (1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC0) is set to 1 for extension code reception and an interrupt request (INTIIC0) is issued at the falling edge of the eighth clock. The local address stored in slave address register 0 (SVA0) is not affected.
- (2) The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer when SVA0 is set to 11110xx0. Note that INTIIC0 occurs at the falling edge of the eighth clock.
  - Higher four bits of data match: EXC0 = 1
  - Seven bits of data match: COI0 = 1

**Remark** EXC0: Bit 5 of IIC status register 0 (IICS0)  
COI0: Bit 4 of IIC status register 0 (IICS0)

- (3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.  
If the extension code is received while a slave device is operating, then the slave device is participating in communication even if its address does not match.  
For example, after the extension code is received, if you do not wish to operate the target device as a slave device, set bit 6 (LREL0) of the IIC control register 0 (IICC0) to 1 to set the standby mode for the next communication operation.

**Table 14-5. Bit Definitions of Major Extension**

Slave Address	R/W Bit	Description
0 0 0 0 0 0 0	0	General call address
1 1 1 1 0 x x	0	10-bit slave address specification (during address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (after address match, when read command is issued)

**Remark** See the I<sup>2</sup>C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

### 14.5.13 Arbitration

When several master devices simultaneously generate a start condition (when STT0 is set to 1 before STD0 is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

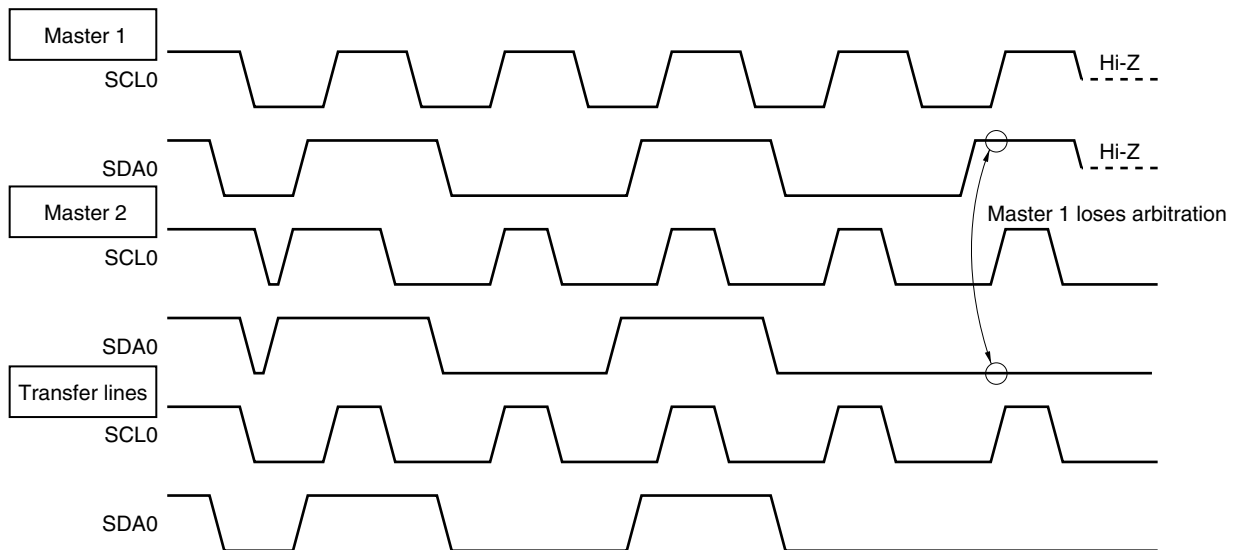
When one of the master devices loses in arbitration, an arbitration loss flag (ALD0) in IIC status register 0 (IICS0) is set (1) via the timing by which the arbitration loss occurred, and the SCL0 and SDA0 lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 = 1 setting that has been made by software.

For details of interrupt request timing, see **14.5.9 Interrupt request (INTIIC0) generation timing and wait control**.

**Remark** STD0: Bit 1 of IIC status register 0 (IICS0)  
STT0: Bit 1 of IIC control register 0 (IICC0)

**Figure 14-20. Arbitration Timing Example**



**Table 14-6. Status During Arbitration and Interrupt Request Generation Timing**

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer <sup>Note 1</sup>
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when SPIE0 = 1) <sup>Note 2</sup>
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer <sup>Note 1</sup>
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIE0 = 1) <sup>Note 2</sup>
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer <sup>Note 1</sup>
When SCL0 is at low level while attempting to generate a restart condition	

**Notes 1.** When WTIM0 (bit 3 of IIC control register 0 (IICC0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0 = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

**2.** When there is a chance that arbitration will occur, set SPIE0 = 1 for master device operation.

**Remark** SPIE0: Bit 4 of IIC control register 0 (IICC0)

#### 14.5.14 Wakeup function

The I<sup>2</sup>C bus slave function is a function that generates an interrupt request signal (INTIIC0) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIIC0 signal from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

However, when a stop condition is detected, bit 4 (SPIE0) of IIC control register 0 (IICC0) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

14.5.15 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IIC flag register 0 (IICF0) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ( $\overline{ACK}$  is not returned and the bus was released when bit 6 (LREL0) of IIC control register 0 (IICC0) was set to 1 and exit from communications).

If bit 1 (STT0) of IICC0 is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.

If an address is written to IIC shift register 0 (IIC0) after bit 4 (SPIE0) of IICC0 was set to 1, and it was detected by generation of an interrupt request signal (INTIIC0) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to IIC0 before the stop condition is detected is invalid.

When STT0 has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released .....a start condition is generated
- If the bus has not been released (standby mode) .....communication reservation

Check whether the communication reservation operates or not by using MSTS0 (bit 7 of IIC status register 0 (IICS0)) after STT0 is set to 1 and the wait time elapses.

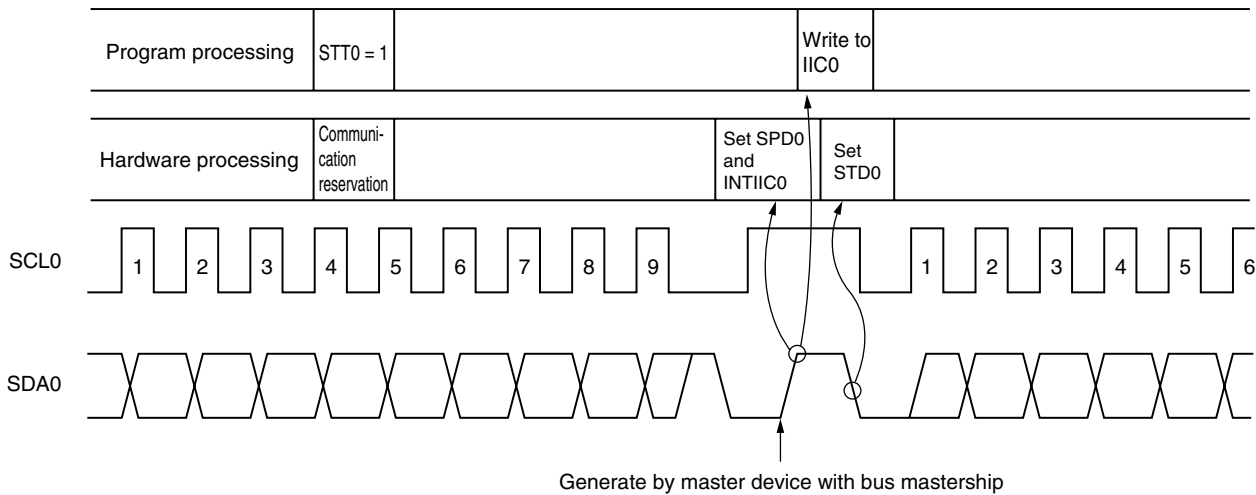
The wait periods, which should be set via software, are listed in Table 14-6.

Table 14-7. Wait Periods

CLX0	SMC0	CL01	CL00	Wait Period
0	0	0	0	43 clocks
0	0	0	1	85 clocks
0	0	1	0	101 clocks
0	0	1	1	23 clocks
0	1	0	0	27 clocks
0	1	0	1	
0	1	1	0	51 clocks
0	1	1	1	15 clocks
1	1	0	0	
1	1	0	1	
1	1	1	0	27 clocks
1	1	1	1	9 clocks

Figure 14-21 shows the communication reservation timing.

**Figure 14-21. Communication Reservation Timing**



**Remark** IIC0: IIC shift register 0  
 STT0: Bit 1 of IIC control register 0 (IICC0)  
 STD0: Bit 1 of IIC status register 0 (IICS0)  
 SPD0: Bit 0 of IIC status register 0 (IICS0)

Communication reservations are accepted via the timing shown in Figure 14-22. After bit 1 (STD0) of IIC status register 0 (IICS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IIC control register 0 (IICC0) to 1 before a stop condition is detected.

**Figure 14-22. Timing for Accepting Communication Reservations**

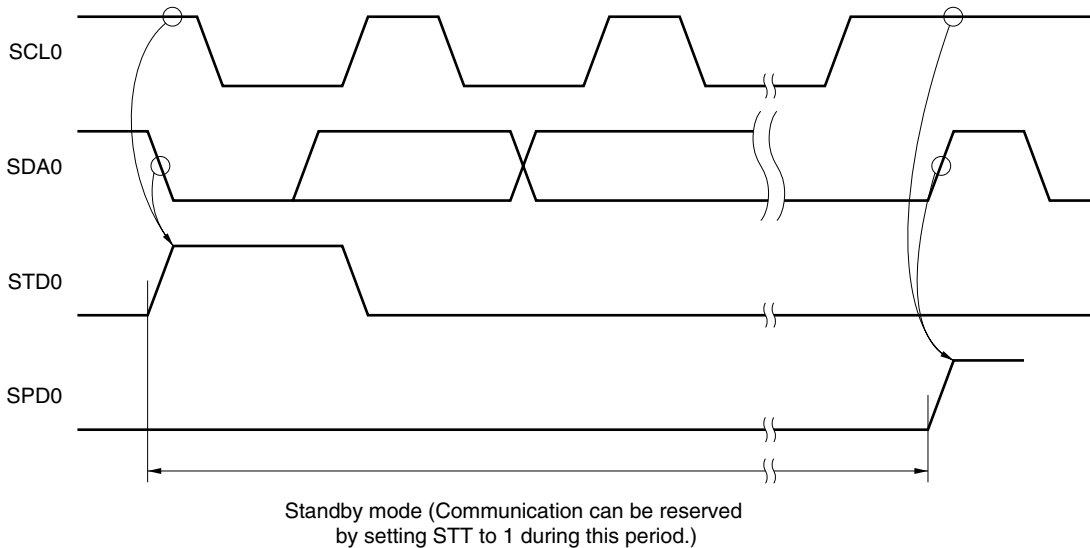
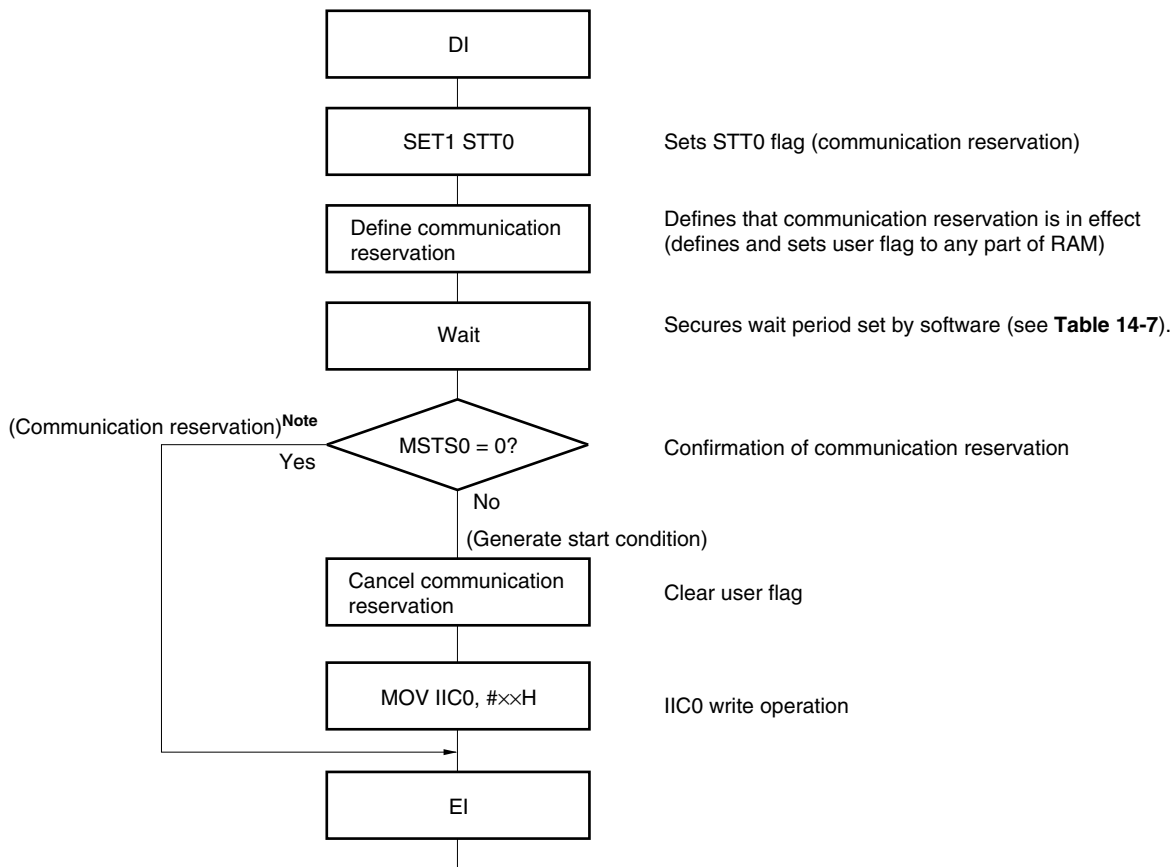


Figure 14-23 shows the communication reservation protocol.

Figure 14-23. Communication Reservation Protocol



**Note** The communication reservation operation executes a write to IIC shift register 0 (IIC0) when a stop condition interrupt request occurs.

**Remark** STT0: Bit 1 of IIC control register 0 (IICC0)  
 MSTS0: Bit 7 of IIC status register 0 (IICS0)  
 IIC0: IIC shift register 0

**(2) When communication reservation function is disabled (bit 0 (IICRSV) of IIC flag register 0 (IICF0) = 1)**

When bit 1 (STT0) of IIC control register 0 (IICC0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ( $\overline{ACK}$  is not returned and the bus was released by setting bit 6 (LREL0) of IICC0 to 1 and exit from communications)

To confirm whether the start condition was generated or request was rejected, check STCF (bit 7 of IICF0). It takes up to 5 clocks until STCF is set to 1 after setting STT0 = 1. Therefore, secure the time by software.



## 14.5.16 Cautions

- (1) When STCEN (bit 1 of IIC flag register 0 (IICF0)) = 0  
 Immediately after I<sup>2</sup>C operation is enabled (IICE0 = 1), the bus communication status (IICBSY (bit 6 of IICF0) = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.  
 When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).  
 Use the following sequence for generating a stop condition.
  - <1> Set IIC clock select register 0 (IICCL0).
  - <2> Set bit 7 (IICE0) of IIC control register 0 (IICC0) to 1.
  - <3> Set bit 0 (SPT0) of IICC0 to 1.
  
- (2) When STCEN = 1  
 Immediately after I<sup>2</sup>C operation is enabled (IICE0 = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT0 (bit 1 of IIC control register 0 (IICC0)) = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.
  
- (3) If other I<sup>2</sup>C communications are already in progress  
 If I<sup>2</sup>C operation is enabled and the device participates in communication already in progress when the SDA0 pin is low and the SCL0 pin is high, the macro of I<sup>2</sup>C recognizes that the SDA0 pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code,  $\overline{ACK}$  is returned, but this interferes with other I<sup>2</sup>C communications. To avoid this, start I<sup>2</sup>C in the following sequence.
  - <1> Clear bit 4 (SPIE0) of IICC0 to 0 to disable generation of an interrupt request signal (INTIIC0) when the stop condition is detected.
  - <2> Set bit 7 (IICE0) of IICC0 to 1 to enable the operation of I<sup>2</sup>C.
  - <3> Wait for detection of the start condition.
  - <4> Set bit 6 (LREL0) of IICC0 to 1 before  $\overline{ACK}$  is returned (4 to 80 clocks after setting IICE0 to 1), to forcibly disable detection.
  
- (4) Determine the transfer clock frequency by using SMC0, CL01, CL00 (bits 3, 1, and 0 of IICL0), and CLX0 (bit 0 of IICX0) before enabling the operation (IICE0 = 1). To change the transfer clock frequency, clear IICE0 to 0 once.
  
- (5) Setting STT0 and SPT0 (bits 1 and 0 of IICC0) again after they are set and before they are cleared to 0 is prohibited.
  
- (6) When transmission is reserved, set SPIE0 (bit 4 of IICL0) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to IIC0 after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set SPIE0 to 1 when MSTS0 (bit 7 of IICS0) is detected by software.

### 14.5.17 Communication operations

The following shows three operation procedures with the flowchart.

#### (1) Master operation in single master system

The flowchart when using the 78K0R/KH3 as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

#### (2) Master operation in multimaster system

In the I<sup>2</sup>C bus multimaster system, whether the bus is released or used cannot be judged by the I<sup>2</sup>C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the 78K0R/KH3 takes part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the 78K0R/KH3 loses in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

#### (3) Slave operation

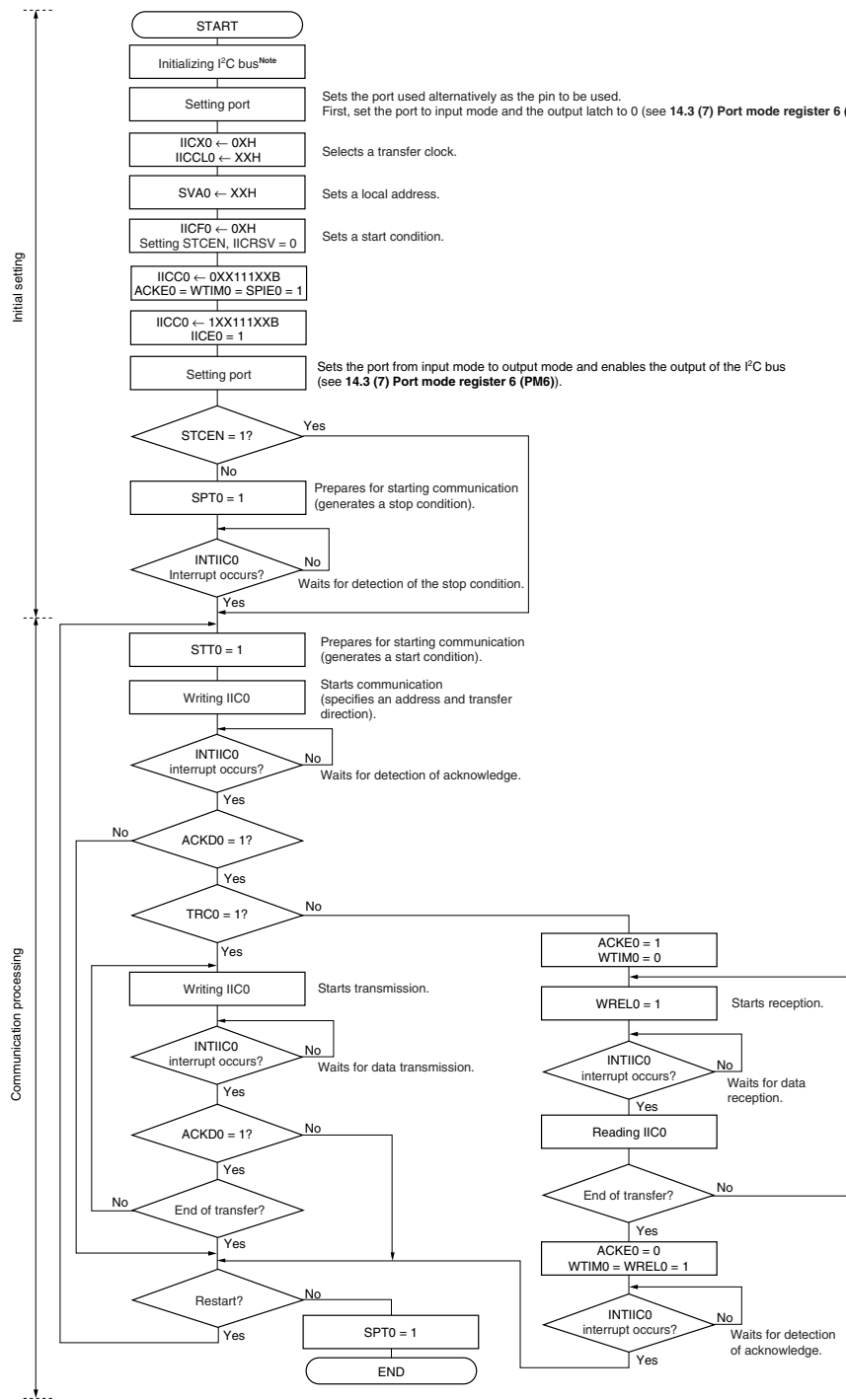
An example of when the 78K0R/KH3 is used as the I<sup>2</sup>C bus slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIIC0 interrupt occurrence (communication waiting). When an INTIIC0 interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

(1) Master operation in single-master system

Figure 14-24. Master Operation in Single-Master System

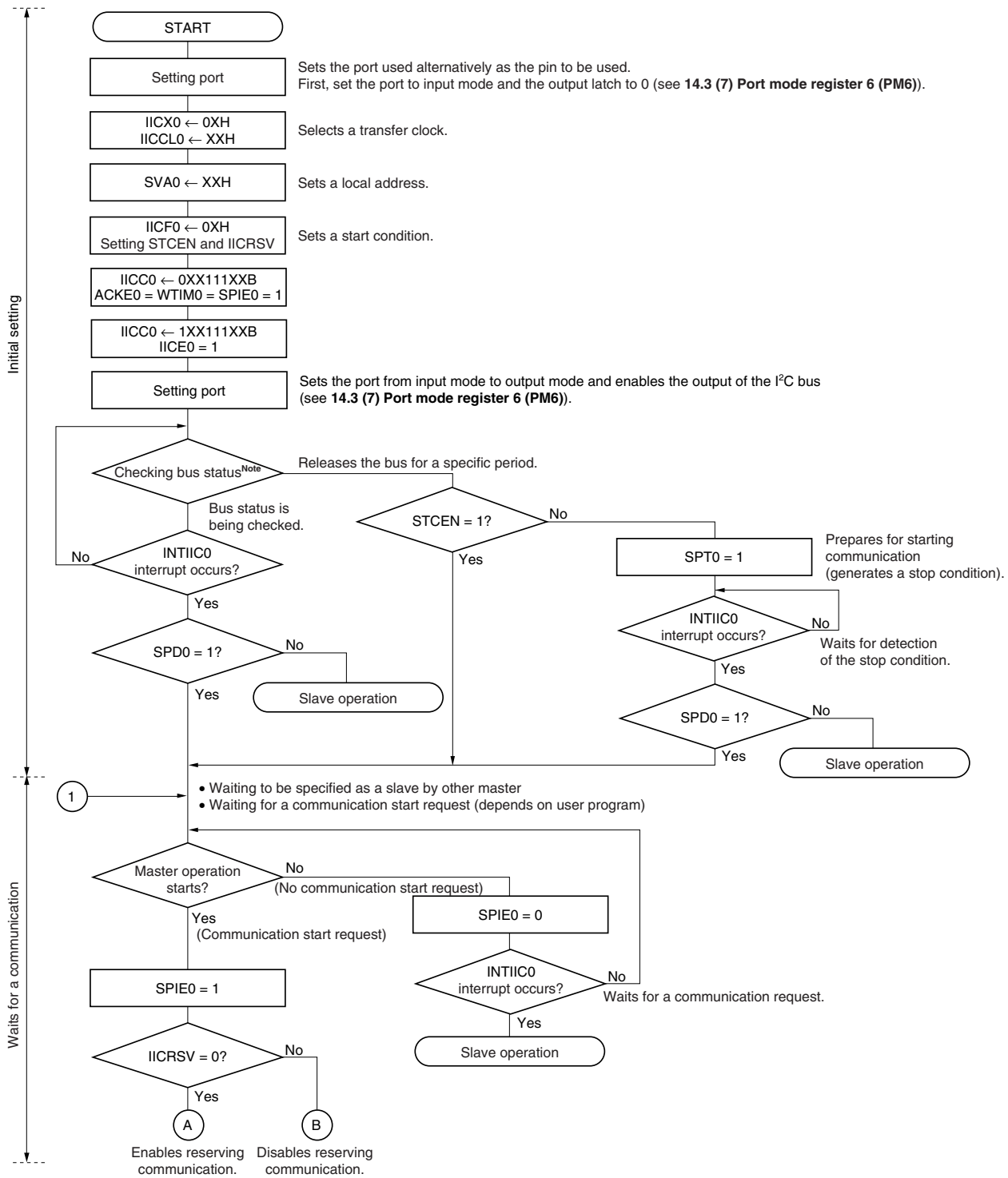


**Note** Release (SCL0 and SDA0 pins = high level) the I<sup>2</sup>C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDA0 pin, for example, set the SCL0 pin in the output port mode, and output a clock pulse from the output port until the SDA0 pin is constantly at high level.

**Remark** Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

(2) Master operation in multi-master system

Figure 14-25. Master Operation in Multi-Master System (1/3)



**Note** Confirm that the bus is released (CLD0 bit = 1, DAD0 bit = 1) for a specific period (for example, for a period of one frame). If the SDA0 pin is constantly at low level, decide whether to release the I<sup>2</sup>C bus (SCL0 and SDA0 pins = high level) in conformance with the specifications of the product that is communicating.

Figure 14-25. Master Operation in Multi-Master System (2/3)

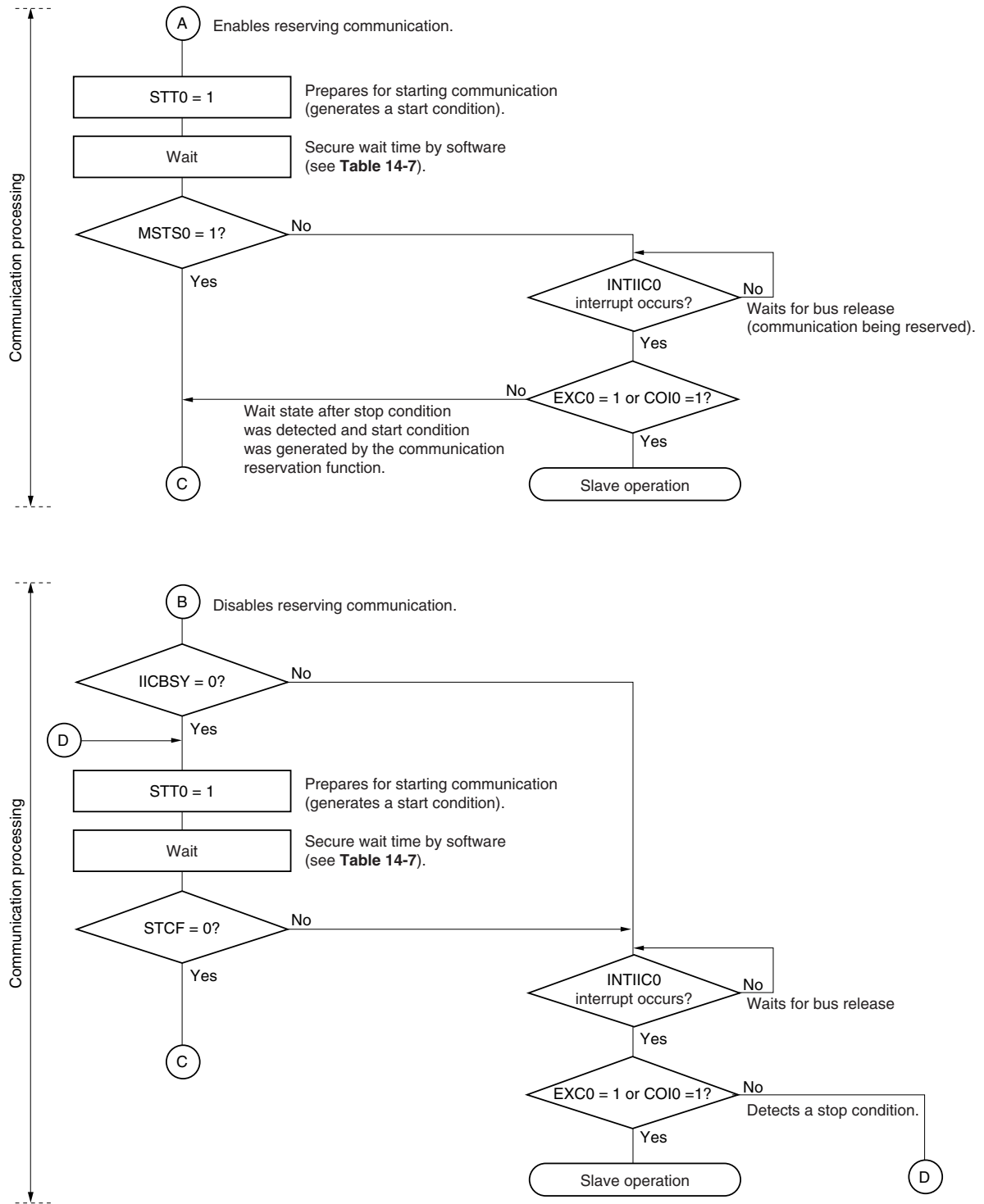
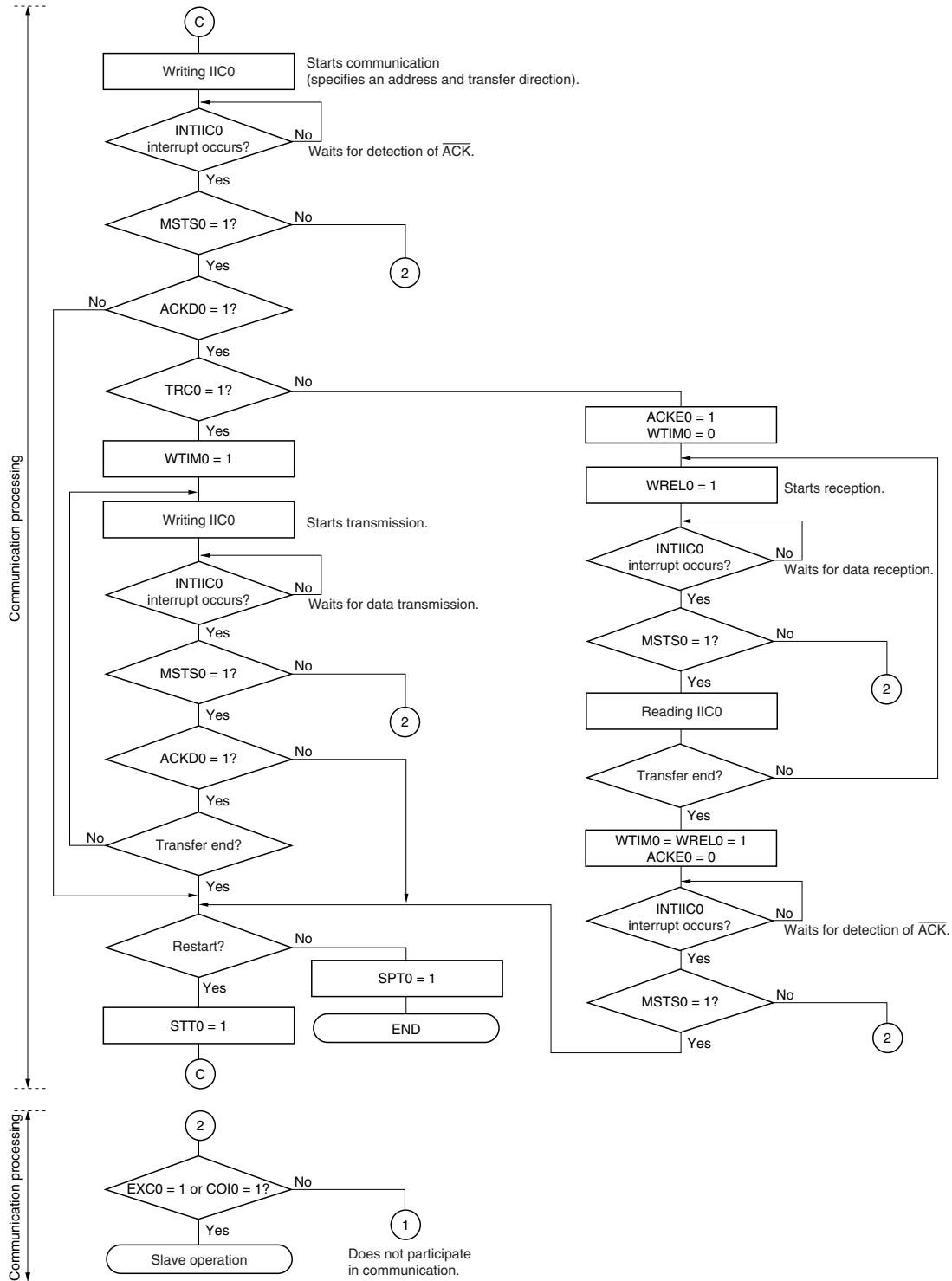


Figure 14-25. Master Operation in Multi-Master System (3/3)



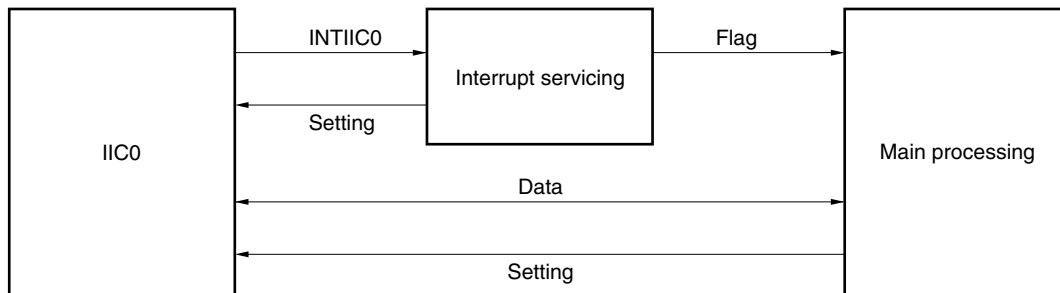
- Remarks**
1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.
  2. To use the device as a master in a multi-master system, read the MSTS0 bit each time interrupt INTIIC0 has occurred to check the arbitration result.
  3. To use the device as a slave in a multi-master system, check the status by using the IICS0 and IICF0 registers each time interrupt INTIIC0 has occurred, and determine the processing to be performed next.

**(3) Slave operation**

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIIC0 interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIIC0 interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIIC0.

**<1> Communication mode flag**

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of ACK from master, address mismatch)

**<2> Ready flag**

This flag indicates that data communication is enabled. Its function is the same as the INTIIC0 interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

**<3> Communication direction flag**

This flag indicates the direction of communication. Its value is the same as TRC0.

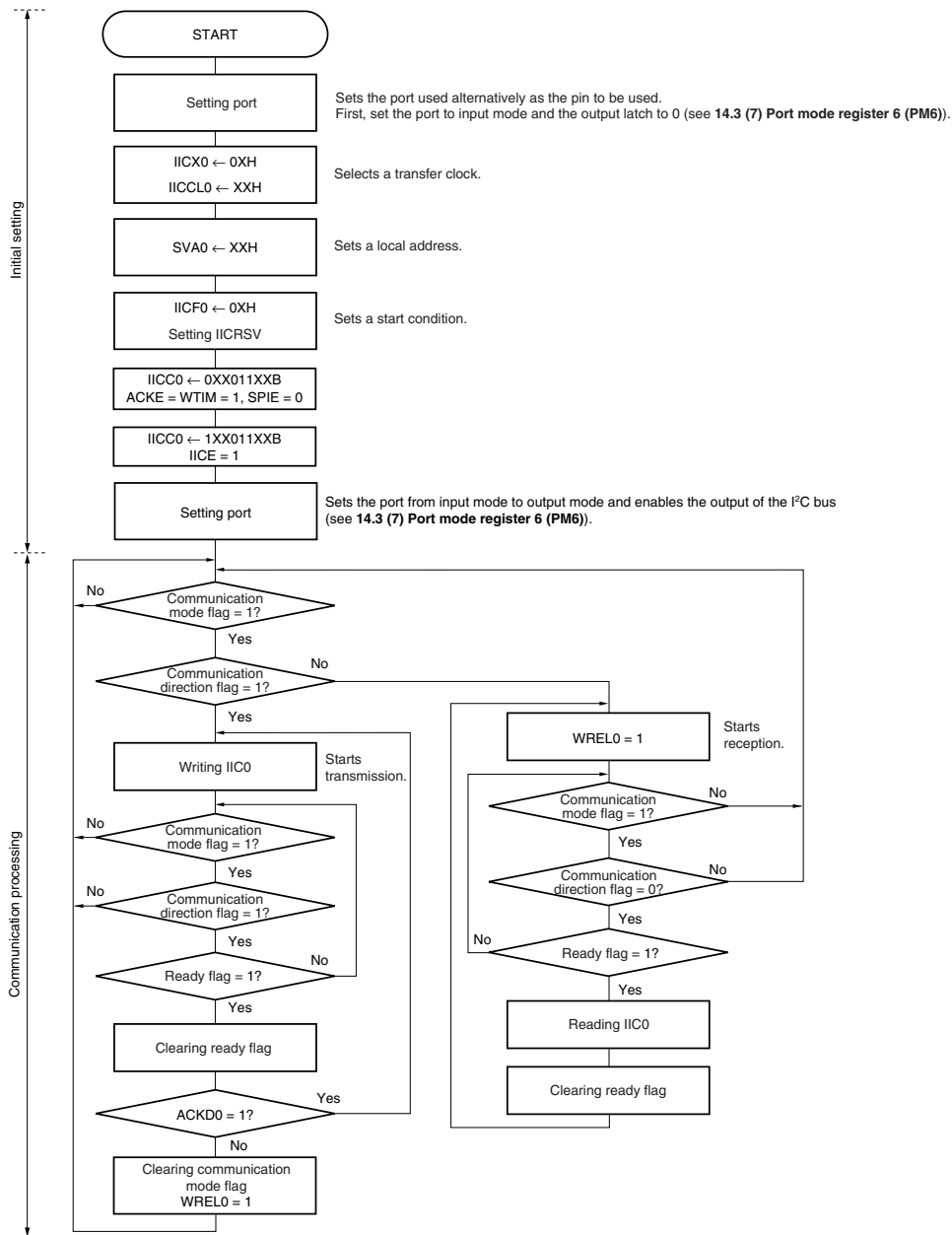
The main processing of the slave operation is explained next.

Start serial interface IIC0 and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns ACK. If  $\overline{\text{ACK}}$  is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed,  $\overline{\text{ACK}}$  is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Figure 14-26. Slave Operation Flowchart (1)



**Remark** Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

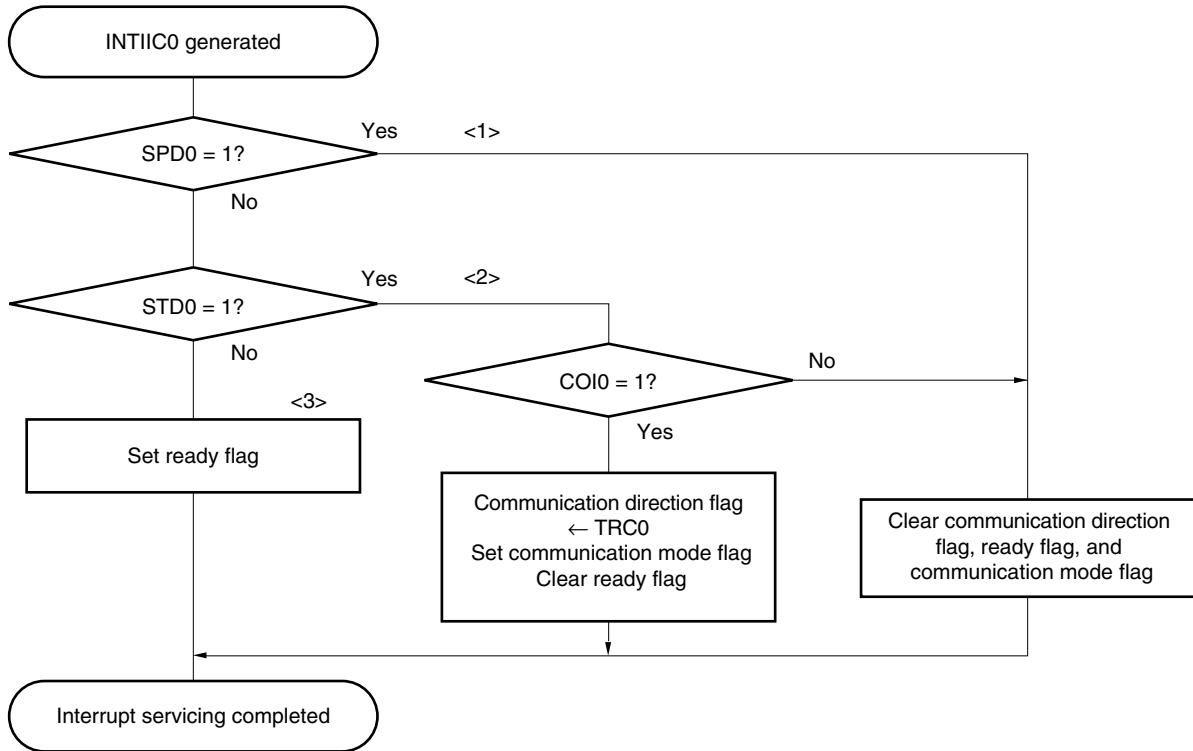


An example of the processing procedure of the slave with the INTIIC0 interrupt is explained below (processing is performed assuming that no extension code is used). The INTIIC0 interrupt checks the status, and the following operations are performed.

- <1> Communication is stopped if the stop condition is issued.
- <2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).
- <3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I<sup>2</sup>C bus remaining in the wait state.

**Remark** <1> to <3> above correspond to <1> to <3> in **Figure 14-27 Slave Operation Flowchart (2)**.

**Figure 14-27. Slave Operation Flowchart (2)**



**14.5.18 Timing of I<sup>2</sup>C interrupt request (INTIIC0) occurrence**

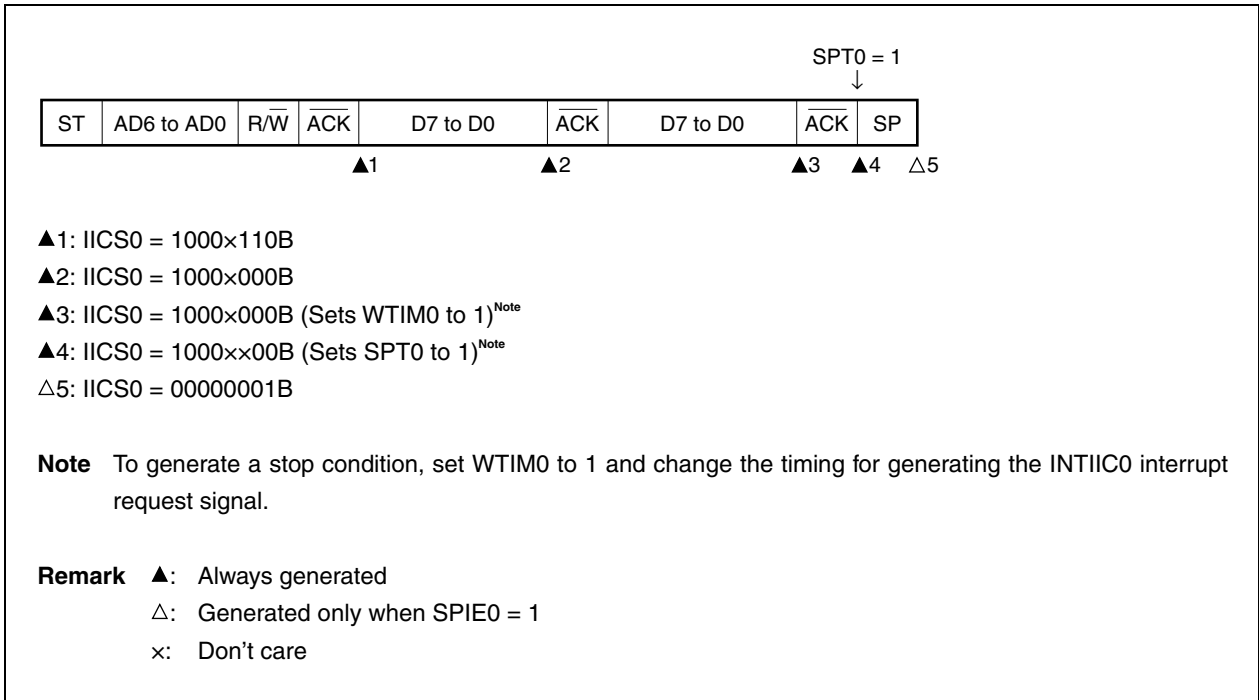
The timing of transmitting or receiving data and generation of interrupt request signal INTIIC0, and the value of the IICS0 register when the INTIIC0 signal is generated are shown below.

<b>Remark</b>	ST:	Start condition
	AD6 to AD0:	Address
	R/W:	Transfer direction specification
	$\overline{\text{ACK}}$ :	Acknowledge
	D7 to D0:	Data
	SP:	Stop condition

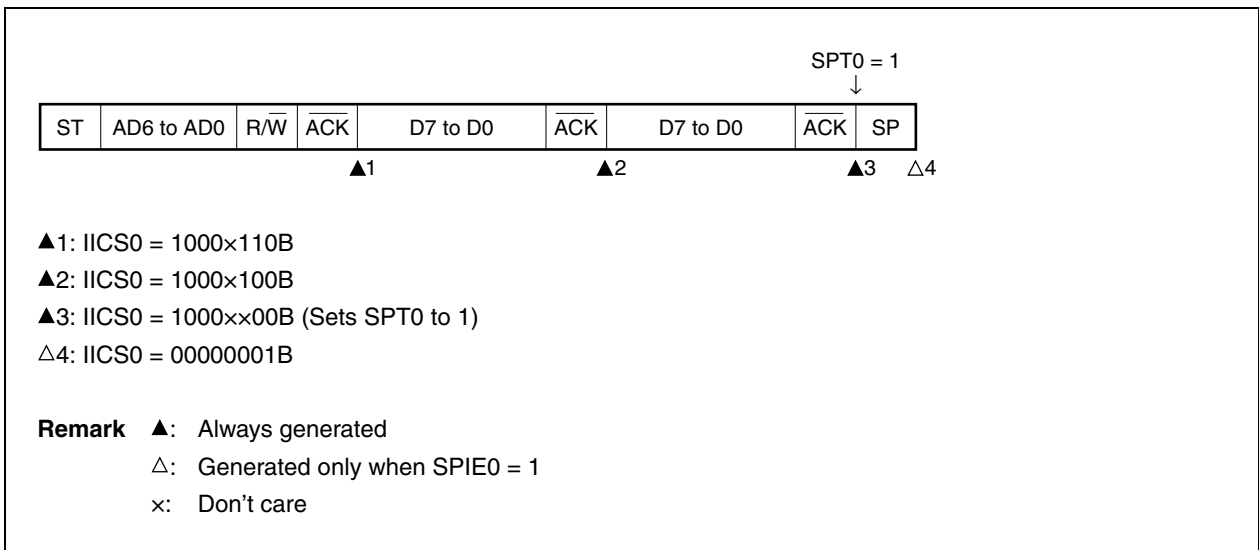
(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM0 = 0

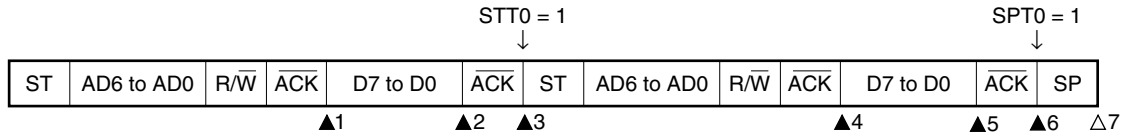


(ii) When WTIM0 = 1



(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0 = 0

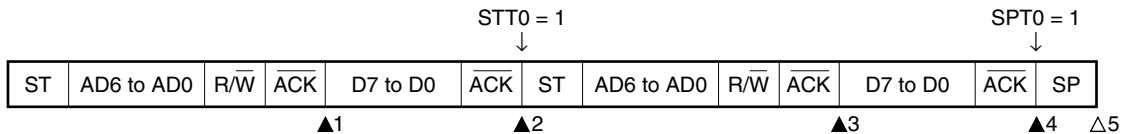


- ▲1: IICS0 = 1000x110B
- ▲2: IICS0 = 1000x000B (Sets WTIM0 to 1)<sup>Note 1</sup>
- ▲3: IICS0 = 1000xx00B (Clears WTIM0 to 0<sup>Note 2</sup>, sets STT0 to 1)
- ▲4: IICS0 = 1000x110B
- ▲5: IICS0 = 1000x000B (Sets WTIM0 to 1)<sup>Note 3</sup>
- ▲6: IICS0 = 1000xx00B (Sets SPT0 to 1)
- Δ7: IICS0 = 00000001B

- Notes 1.** To generate a start condition, set WTIM0 to 1 and change the timing for generating the INTIIC0 interrupt request signal.
- 2.** Clear WTIM0 to 0 to restore the original setting.
- 3.** To generate a stop condition, set WTIM0 to 1 and change the timing for generating the INTIIC0 interrupt request signal.

**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 = 1  
 x: Don't care

(ii) When WTIM0 = 1

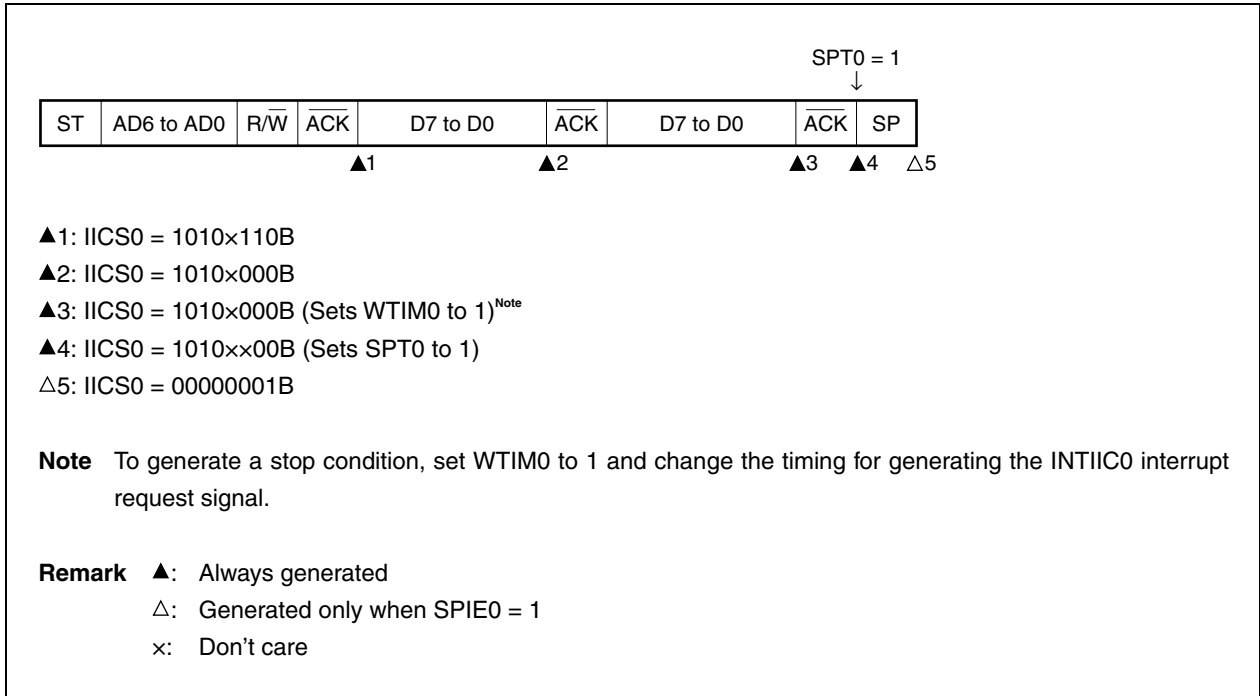


- ▲1: IICS0 = 1000x110B
- ▲2: IICS0 = 1000xx00B (Sets STT0 to 1)
- ▲3: IICS0 = 1000x110B
- ▲4: IICS0 = 1000xx00B (Sets SPT0 to 1)
- Δ5: IICS0 = 00000001B

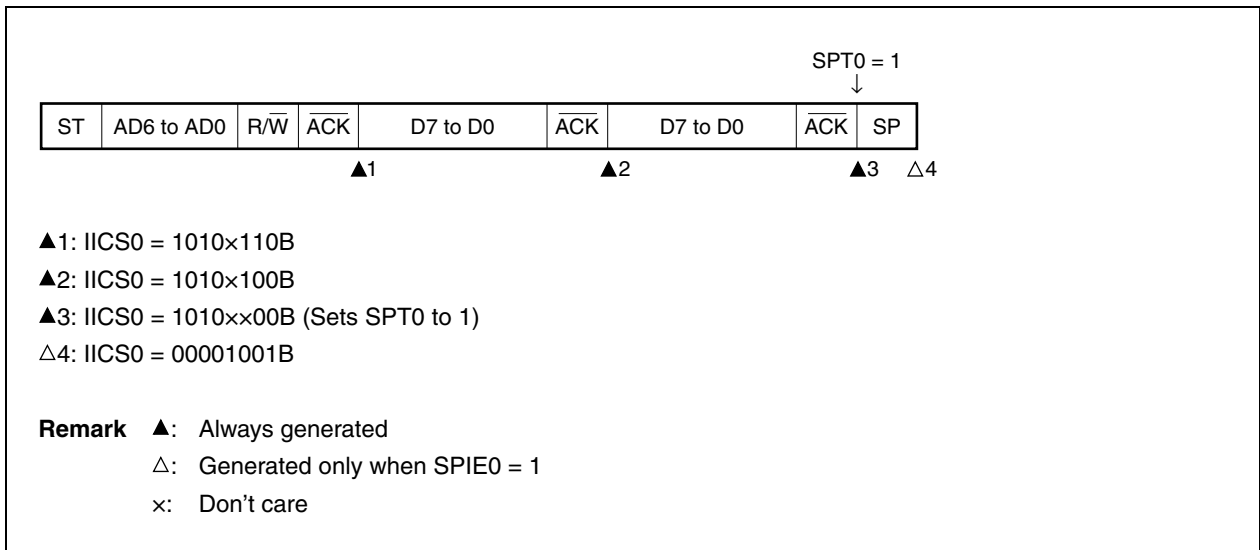
**Remark** ▲: Always generated  
 Δ: Generated only when SPIE0 = 1  
 x: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0



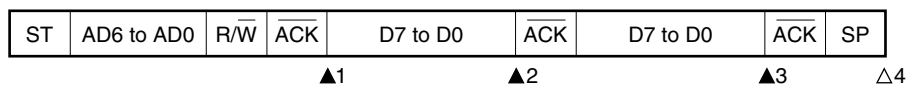
(ii) When WTIM0 = 1



## (2) Slave device operation (slave address data reception)

## (a) Start ~ Address ~ Data ~ Data ~ Stop

## (i) When WTIM0 = 0



▲1: IICS0 = 0001x110B

▲2: IICS0 = 0001x000B

▲3: IICS0 = 0001x000B

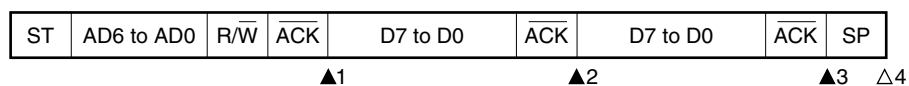
△4: IICS0 = 00000001B

**Remark** ▲: Always generated

△: Generated only when SPIE0 = 1

x: Don't care

## (ii) When WTIM0 = 1



▲1: IICS0 = 0001x110B

▲2: IICS0 = 0001x100B

▲3: IICS0 = 0001xx00B

△4: IICS0 = 00000001B

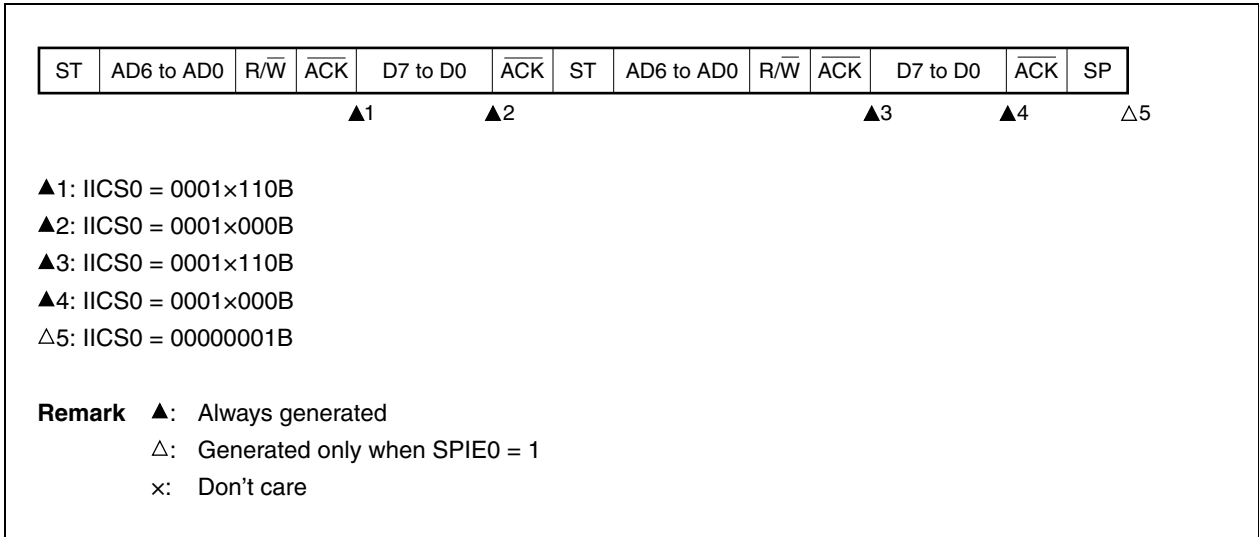
**Remark** ▲: Always generated

△: Generated only when SPIE0 = 1

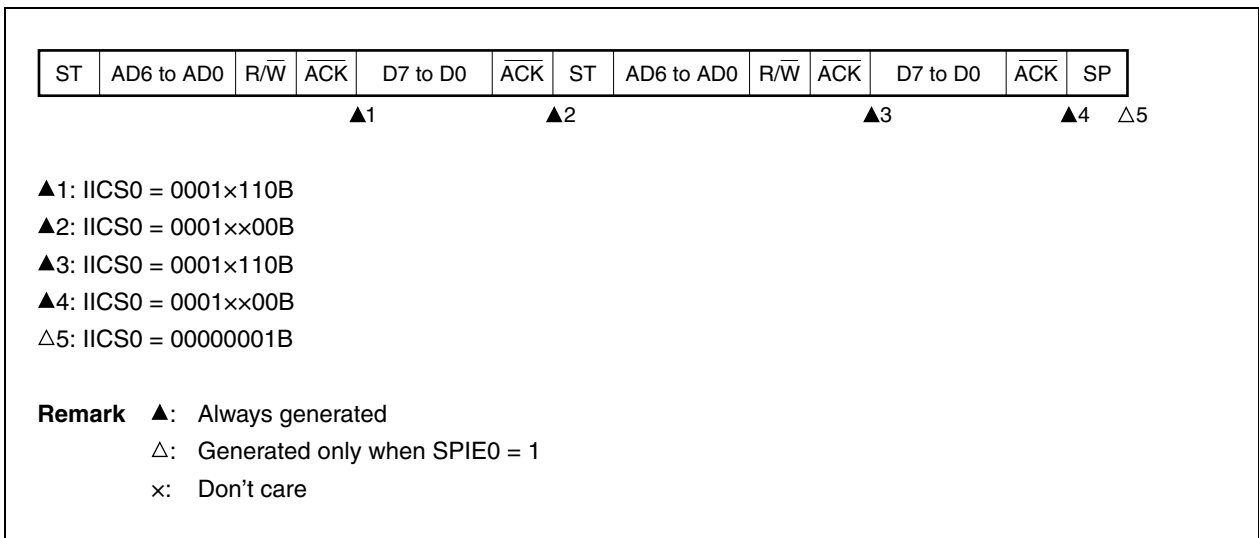
x: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, matches with SVA0)

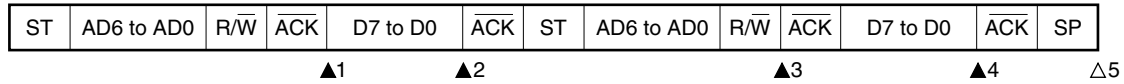


(ii) When WTIM0 = 1 (after restart, matches with SVA0)



(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

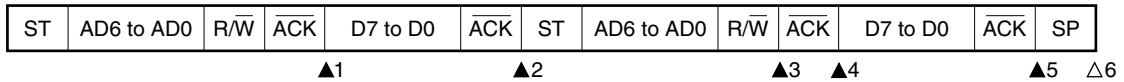
(i) When WTIM0 = 0 (after restart, does not match address (= extension code))



- ▲1: IICS0 = 0001x110B
- ▲2: IICS0 = 0001x000B
- ▲3: IICS0 = 0010x010B
- ▲4: IICS0 = 0010x000B
- △5: IICS0 = 00000001B

**Remark** ▲: Always generated  
 △: Generated only when SPIE0 = 1  
 x: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= extension code))



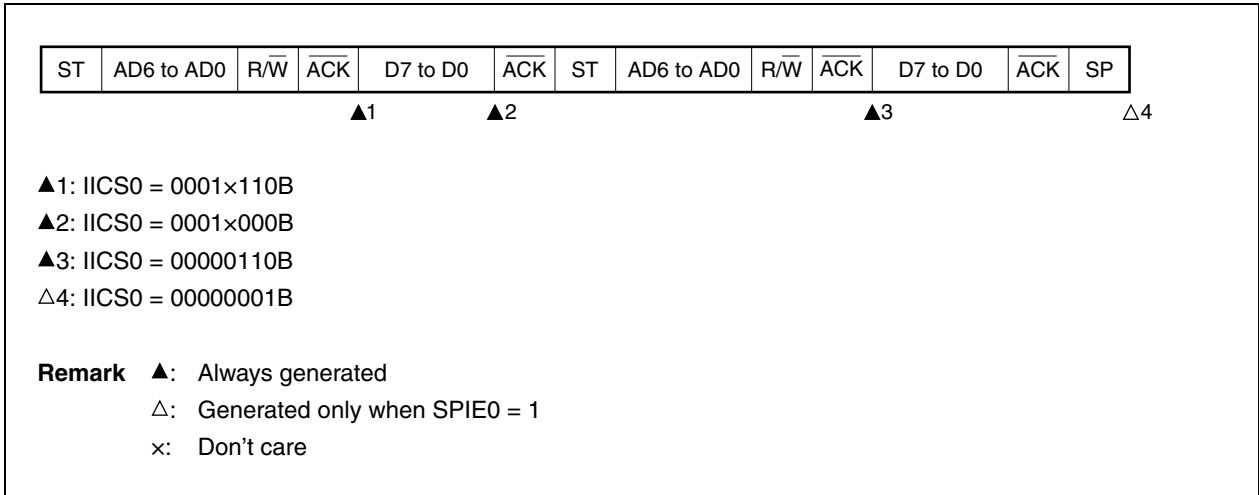
- ▲1: IICS0 = 0001x110B
- ▲2: IICS0 = 0001xx00B
- ▲3: IICS0 = 0010x010B
- ▲4: IICS0 = 0010x110B
- ▲5: IICS0 = 0010xx00B
- △6: IICS0 = 00000001B

**Remark** ▲: Always generated  
 △: Generated only when SPIE0 = 1  
 x: Don't care

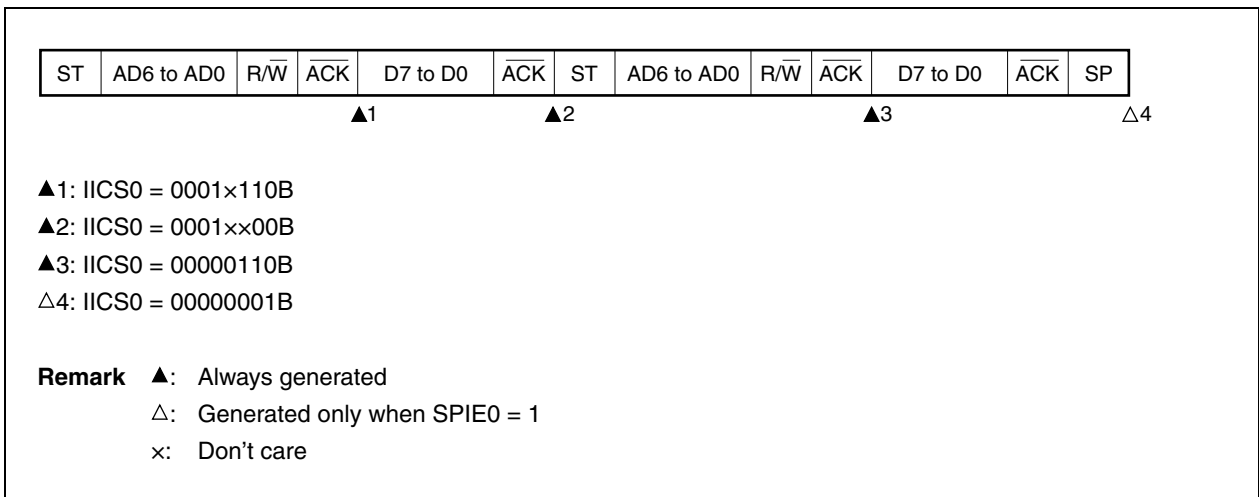


(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))



(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))

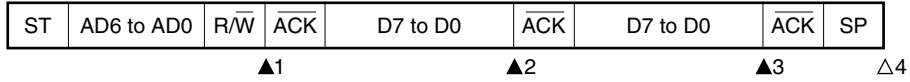


**(3) Slave device operation (when receiving extension code)**

The device is always participating in communication when it receives an extension code.

**(a) Start ~ Code ~ Data ~ Data ~ Stop**

**(i) When WTIM0 = 0**



▲1: IICS0 = 0010x010B

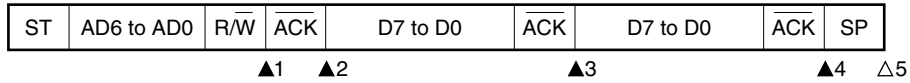
▲2: IICS0 = 0010x000B

▲3: IICS0 = 0010x000B

△4: IICS0 = 00000001B

**Remark** ▲: Always generated  
 △: Generated only when SPIE0 = 1  
 x: Don't care

**(ii) When WTIM0 = 1**



▲1: IICS0 = 0010x010B

▲2: IICS0 = 0010x110B

▲3: IICS0 = 0010x100B

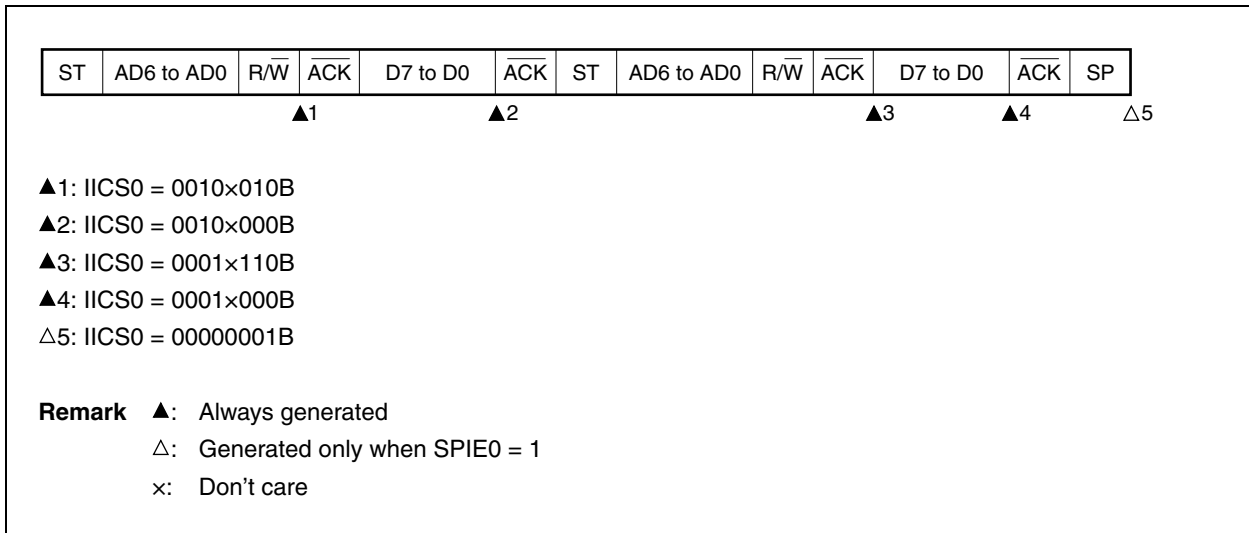
▲4: IICS0 = 0010xx00B

△5: IICS0 = 00000001B

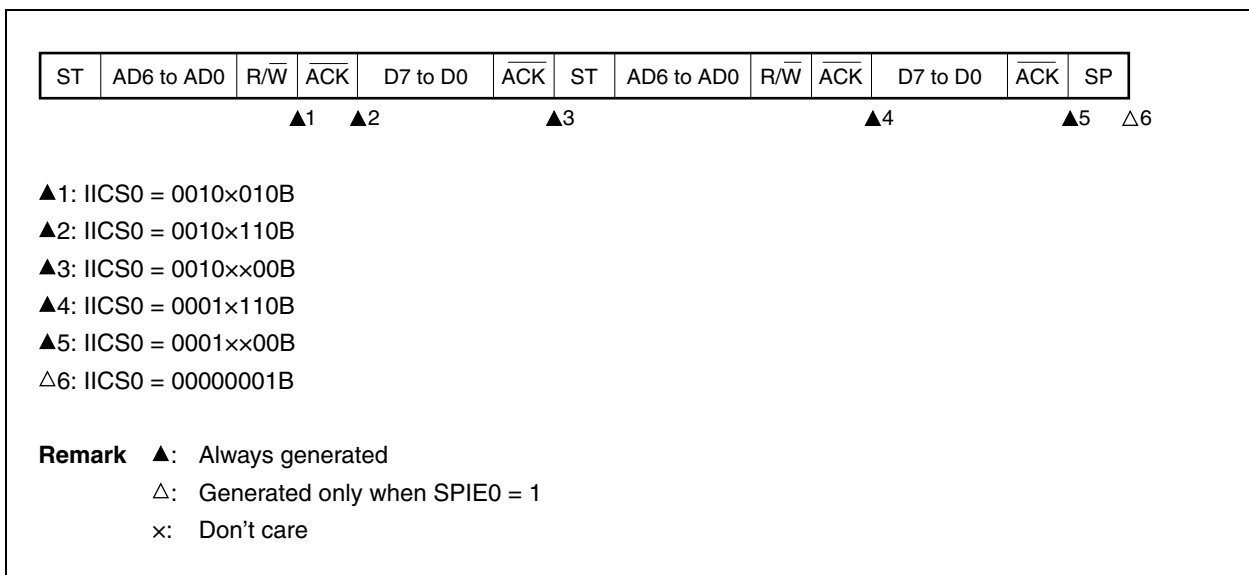
**Remark** ▲: Always generated  
 △: Generated only when SPIE0 = 1  
 x: Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, matches SVA0)

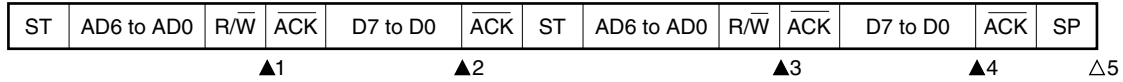


(ii) When WTIM0 = 1 (after restart, matches SVA0)



(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

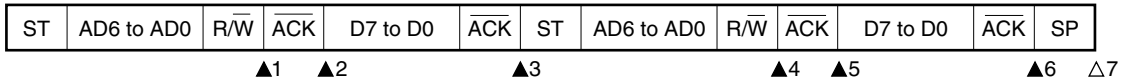
(i) When WTIM0 = 0 (after restart, extension code reception)



- ▲1: IICS0 = 0010x010B
- ▲2: IICS0 = 0010x000B
- ▲3: IICS0 = 0010x010B
- ▲4: IICS0 = 0010x000B
- △5: IICS0 = 00000001B

**Remark** ▲: Always generated  
 △: Generated only when SPIE0 = 1  
 x: Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)

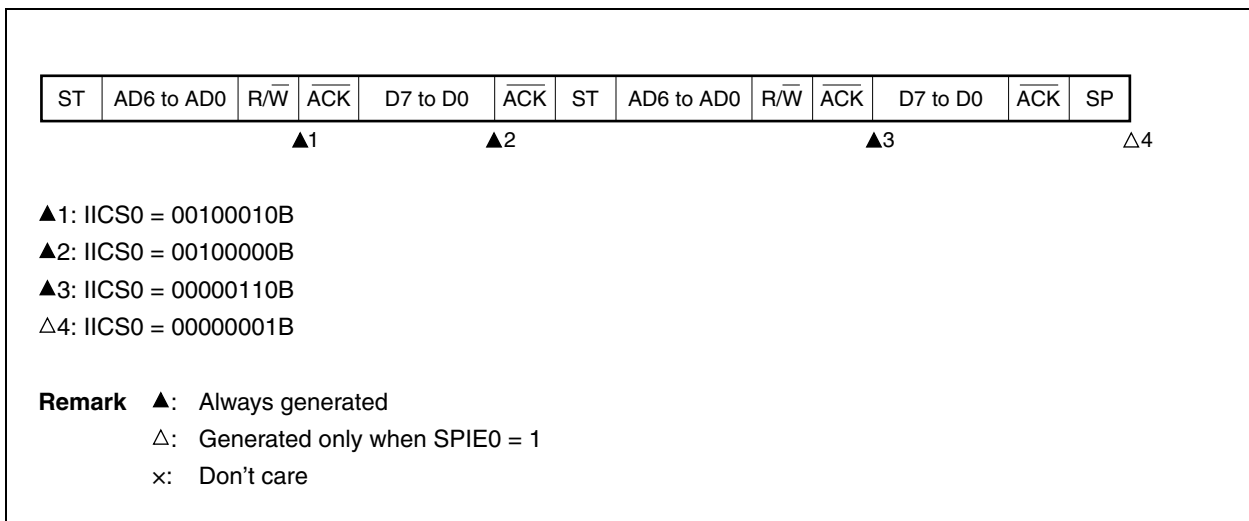


- ▲1: IICS0 = 0010x010B
- ▲2: IICS0 = 0010x110B
- ▲3: IICS0 = 0010xx00B
- ▲4: IICS0 = 0010x010B
- ▲5: IICS0 = 0010x110B
- ▲6: IICS0 = 0010xx00B
- △7: IICS0 = 00000001B

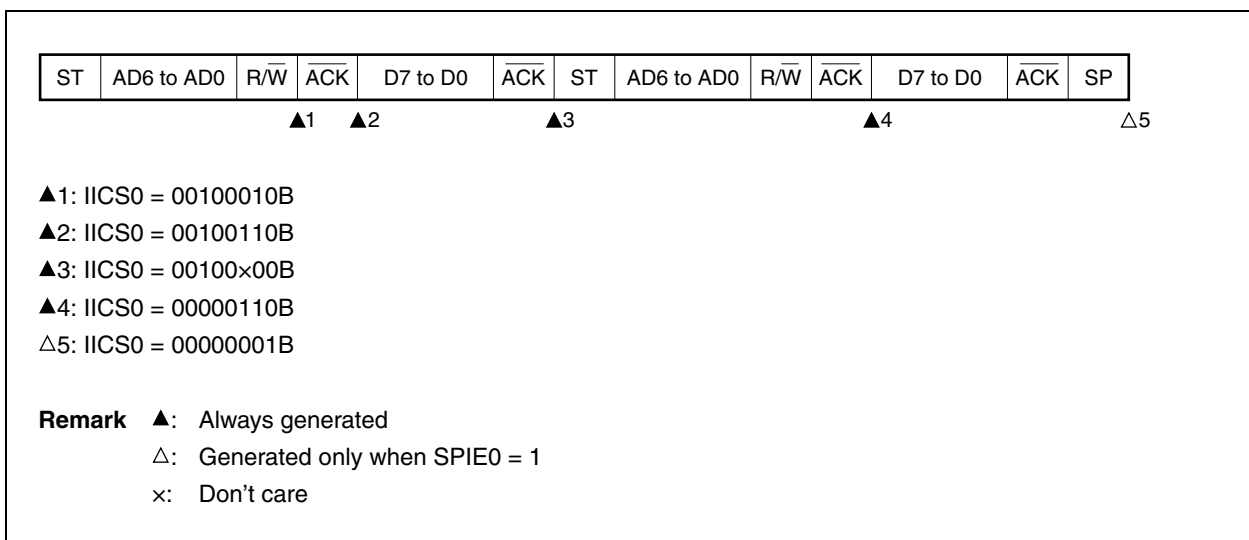
**Remark** ▲: Always generated  
 △: Generated only when SPIE0 = 1  
 x: Don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))

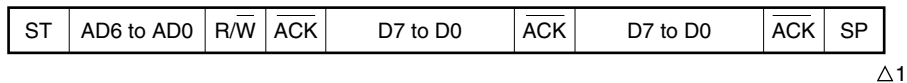


(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))



**(4) Operation without communication**

**(a) Start ~ Code ~ Data ~ Data ~ Stop**



△1: IIC0 = 00000001B

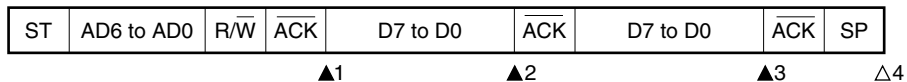
**Remark** △: Generated only when SPIE0 = 1

**(5) Arbitration loss operation (operation as slave after arbitration loss)**

When the device is used as a master in a multi-master system, read the MSTSO bit each time interrupt request signal INTIIC0 has occurred to check the arbitration result.

**(a) When arbitration loss occurs during transmission of slave address data**

**(i) When WTIM0 = 0**



▲1: IIC0 = 0101x110B

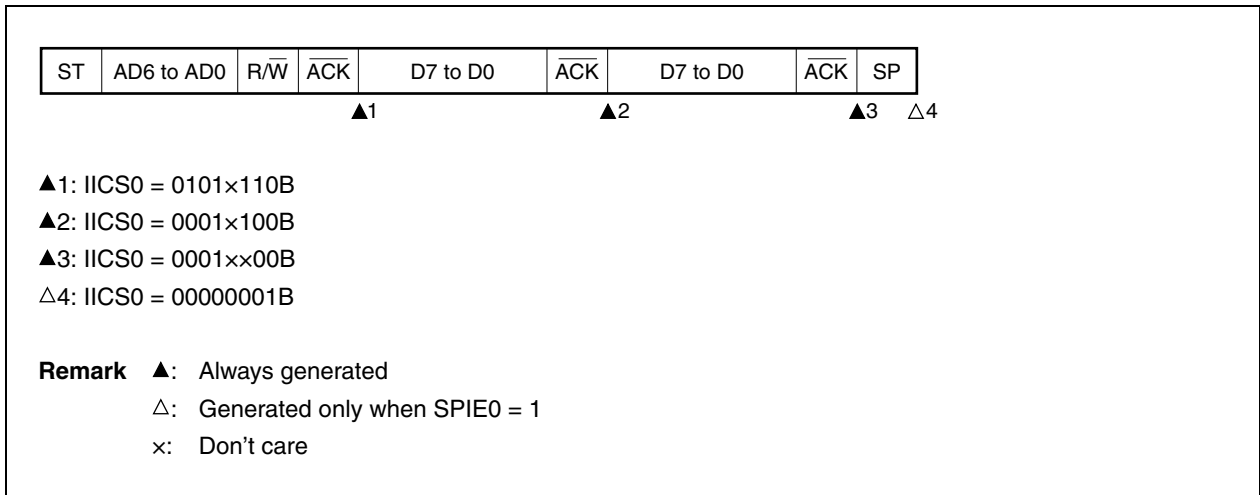
▲2: IIC0 = 0001x000B

▲3: IIC0 = 0001x000B

△4: IIC0 = 00000001B

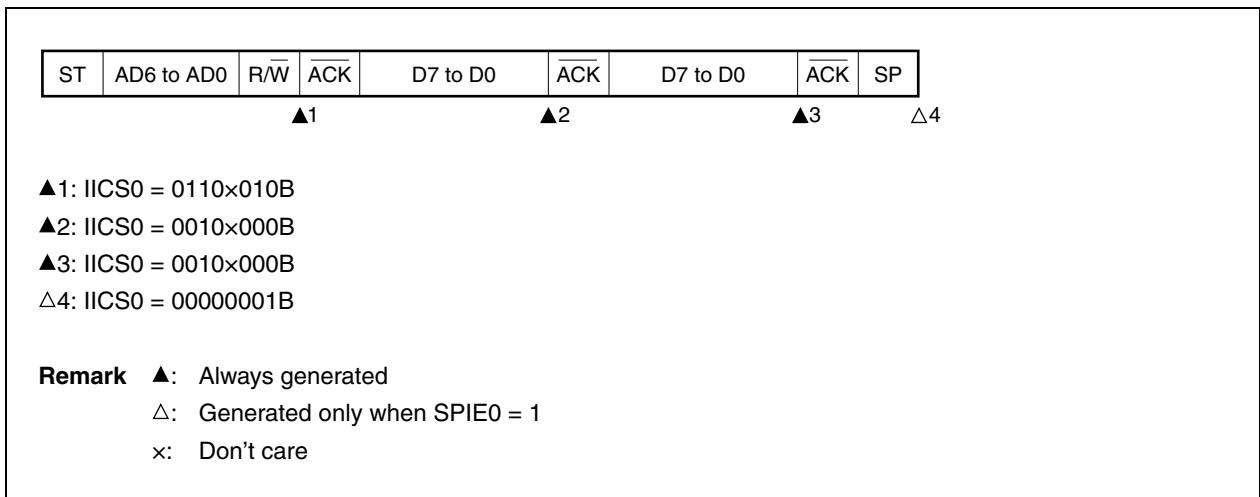
**Remark** ▲: Always generated  
 △: Generated only when SPIE0 = 1  
 x: Don't care

(ii) When  $WTIM0 = 1$

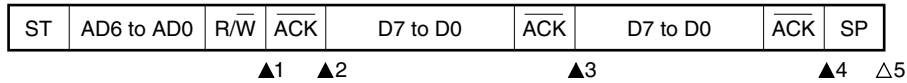


(b) When arbitration loss occurs during transmission of extension code

(i) When  $WTIM0 = 0$



## (ii) When WTIM0 = 1



▲1: IICS0 = 0110x010B

▲2: IICS0 = 0010x110B

▲3: IICS0 = 0010x100B

▲4: IICS0 = 0010xx00B

△5: IICS0 = 00000001B

**Remark** ▲: Always generated

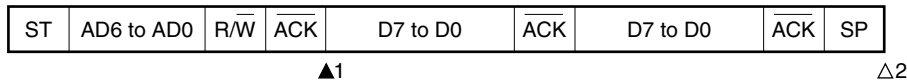
△: Generated only when SPIE0 = 1

x: Don't care

## (6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTSO bit each time interrupt request signal INTIIC0 has occurred to check the arbitration result.

## (a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)



▲1: IICS0 = 01000110B

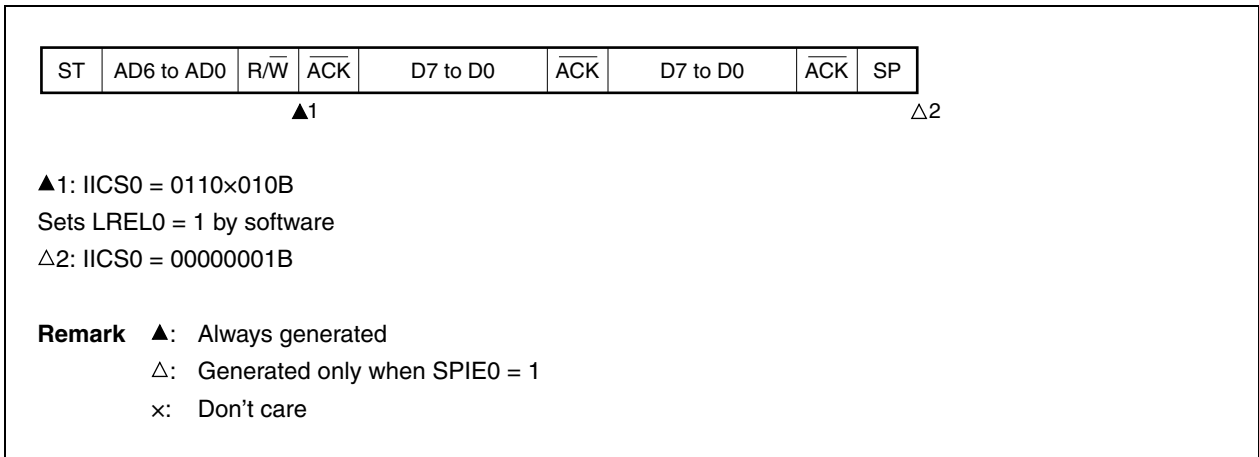
△2: IICS0 = 00000001B

**Remark** ▲: Always generated

△: Generated only when SPIE0 = 1

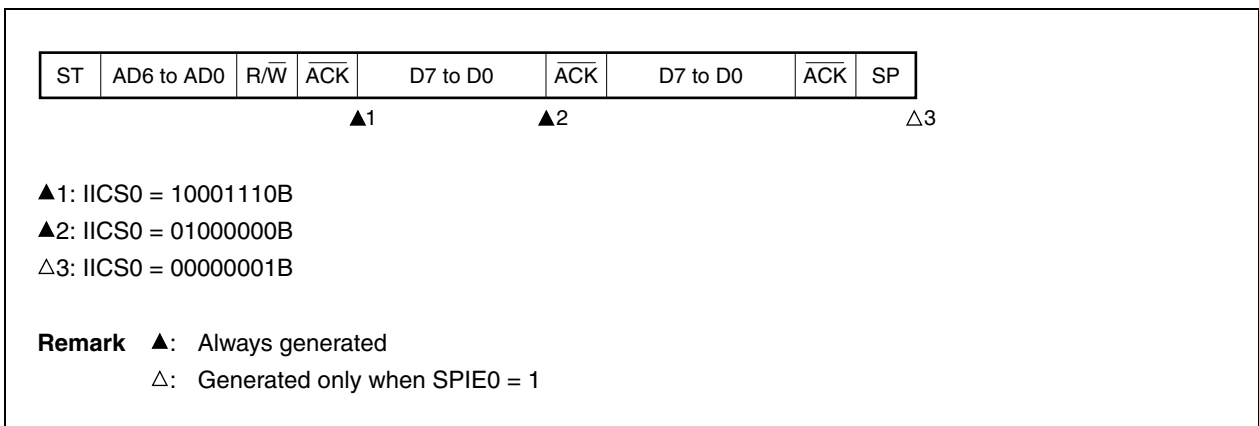


**(b) When arbitration loss occurs during transmission of extension code**

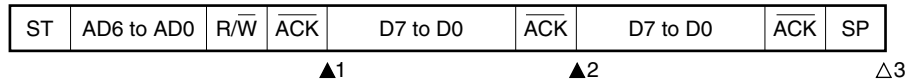


**(c) When arbitration loss occurs during transmission of data**

**(i) When WTIM0 = 0**



(ii) When WTIM0 = 1



▲1: IICS0 = 10001110B

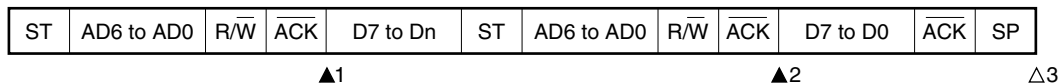
▲2: IICS0 = 01000100B

△3: IICS0 = 00000001B

**Remark** ▲: Always generated  
 △: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatched with SVA0)



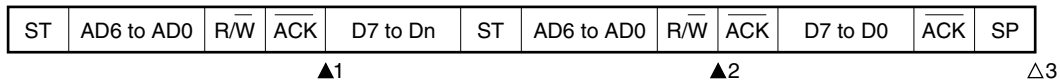
▲1: IICS0 = 1000x110B

▲2: IICS0 = 01000110B

△3: IICS0 = 00000001B

**Remark** ▲: Always generated  
 △: Generated only when SPIE0 = 1  
 x: Don't care  
 n = 6 to 0

(ii) Extension code



▲1: IICS0 = 1000x110B

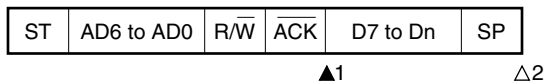
▲2: IICS0 = 01100010B

Sets LREL0 = 1 by software

△3: IICS0 = 00000001B

**Remark** ▲: Always generated  
 △: Generated only when SPIE0 = 1  
 x: Don't care  
 n = 6 to 0

(e) When loss occurs due to stop condition during data transfer



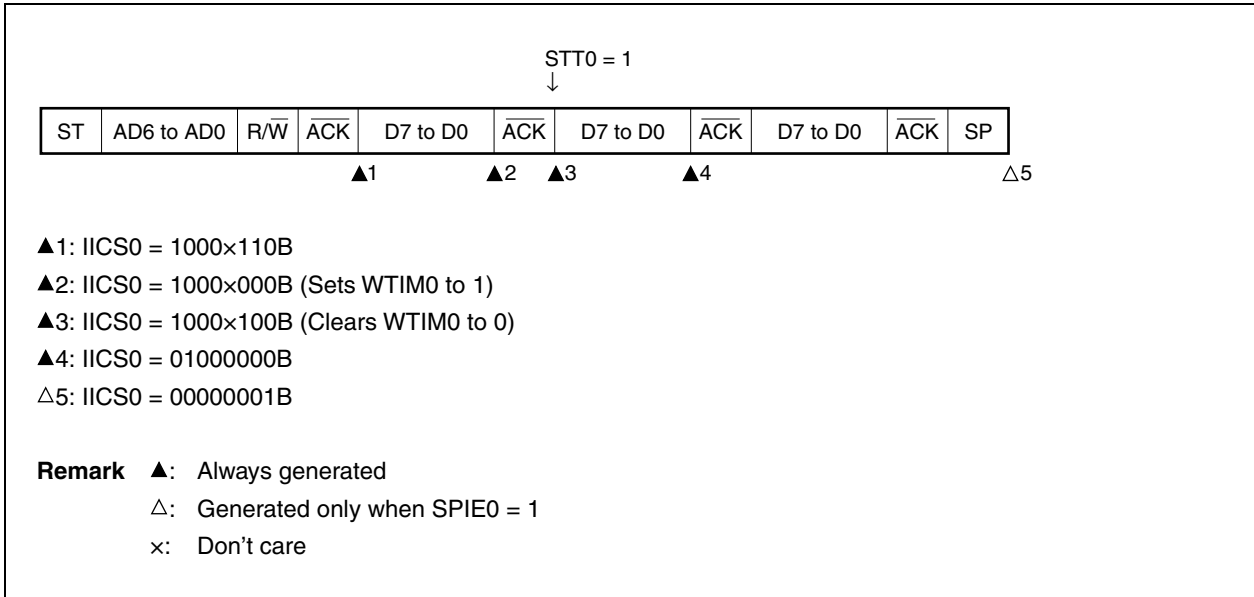
▲1: IICS0 = 10000110B

△2: IICS0 = 01000001B

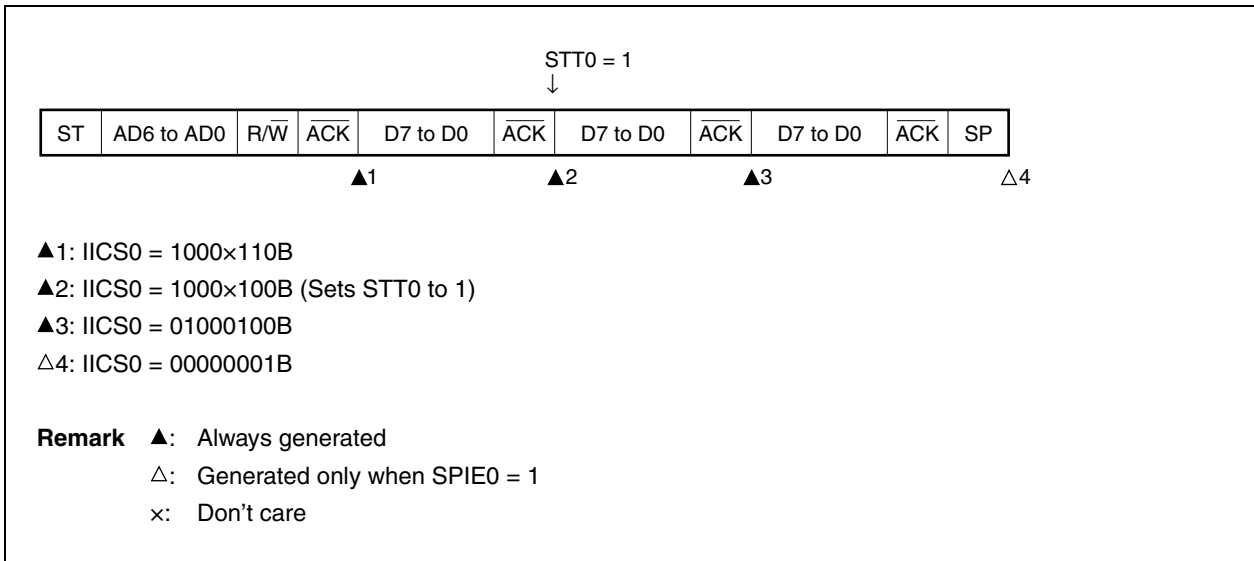
**Remark** ▲: Always generated  
 △: Generated only when SPIE0 = 1  
 x: Don't care  
 n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

(i) When WTIM0 = 0

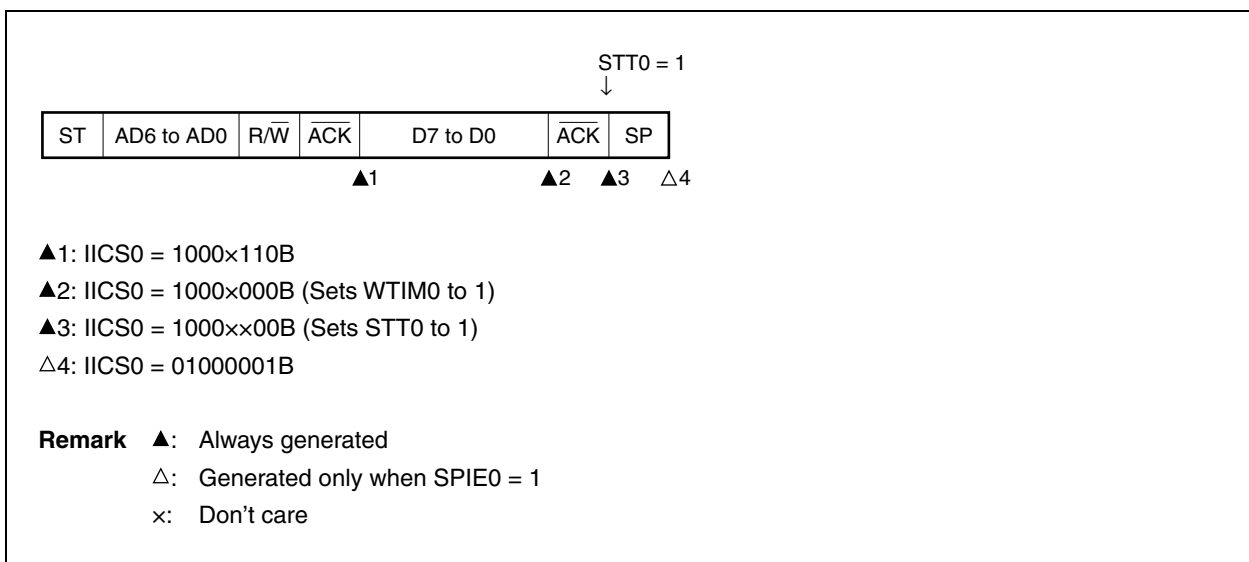


(ii) When WTIM0 = 1

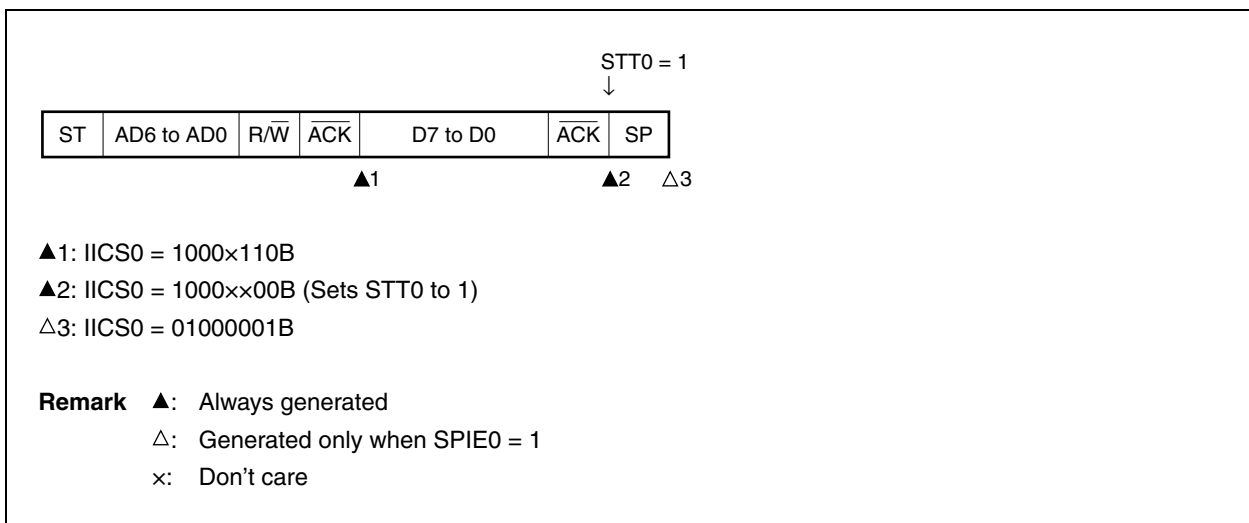


(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

(i) When  $WTIM0 = 0$

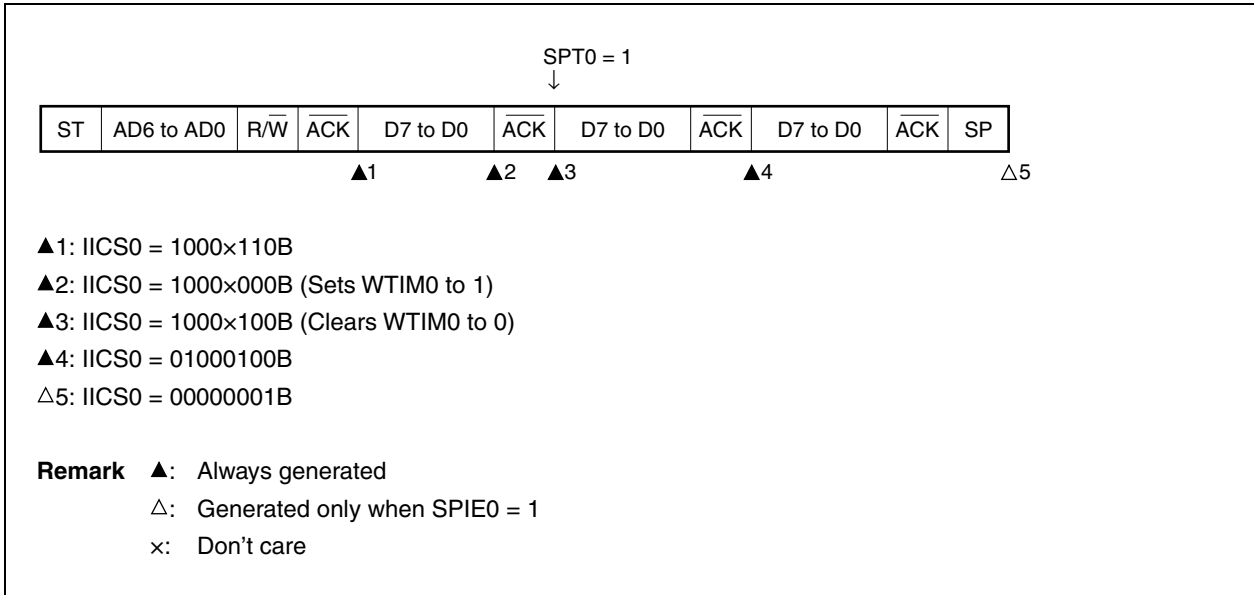


(ii) When  $WTIM0 = 1$

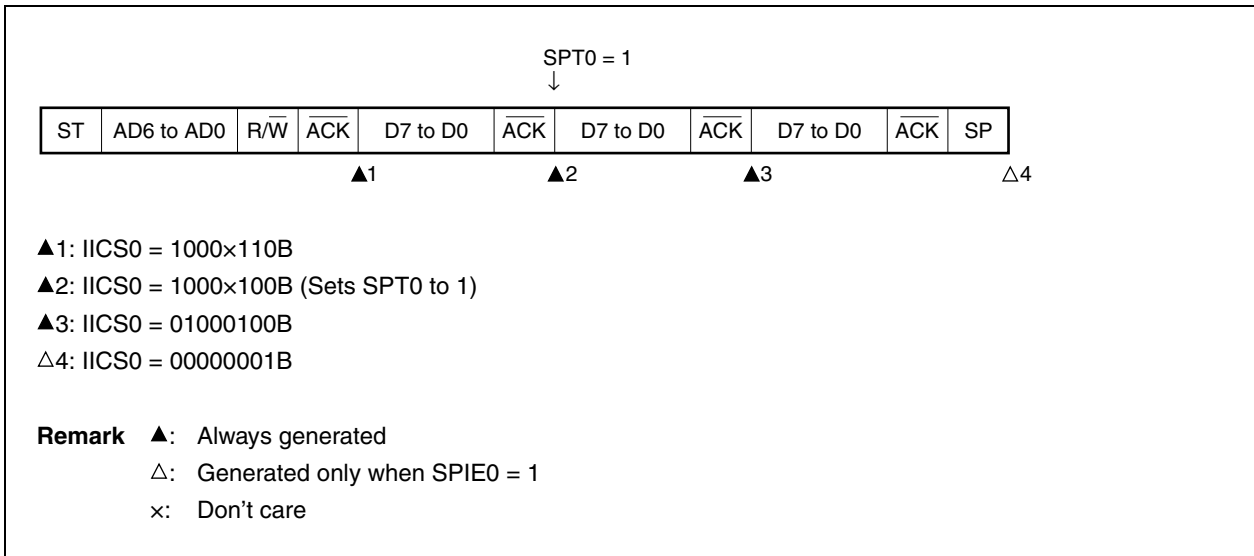


(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

(i) When  $WTIM0 = 0$



(ii) When  $WTIM0 = 1$



## 14.6 Timing Charts

When using the I<sup>2</sup>C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC0 bit (bit 3 of IIC status register 0 (IICS0)), which specifies the data transfer direction, and then starts serial communication with the slave device.

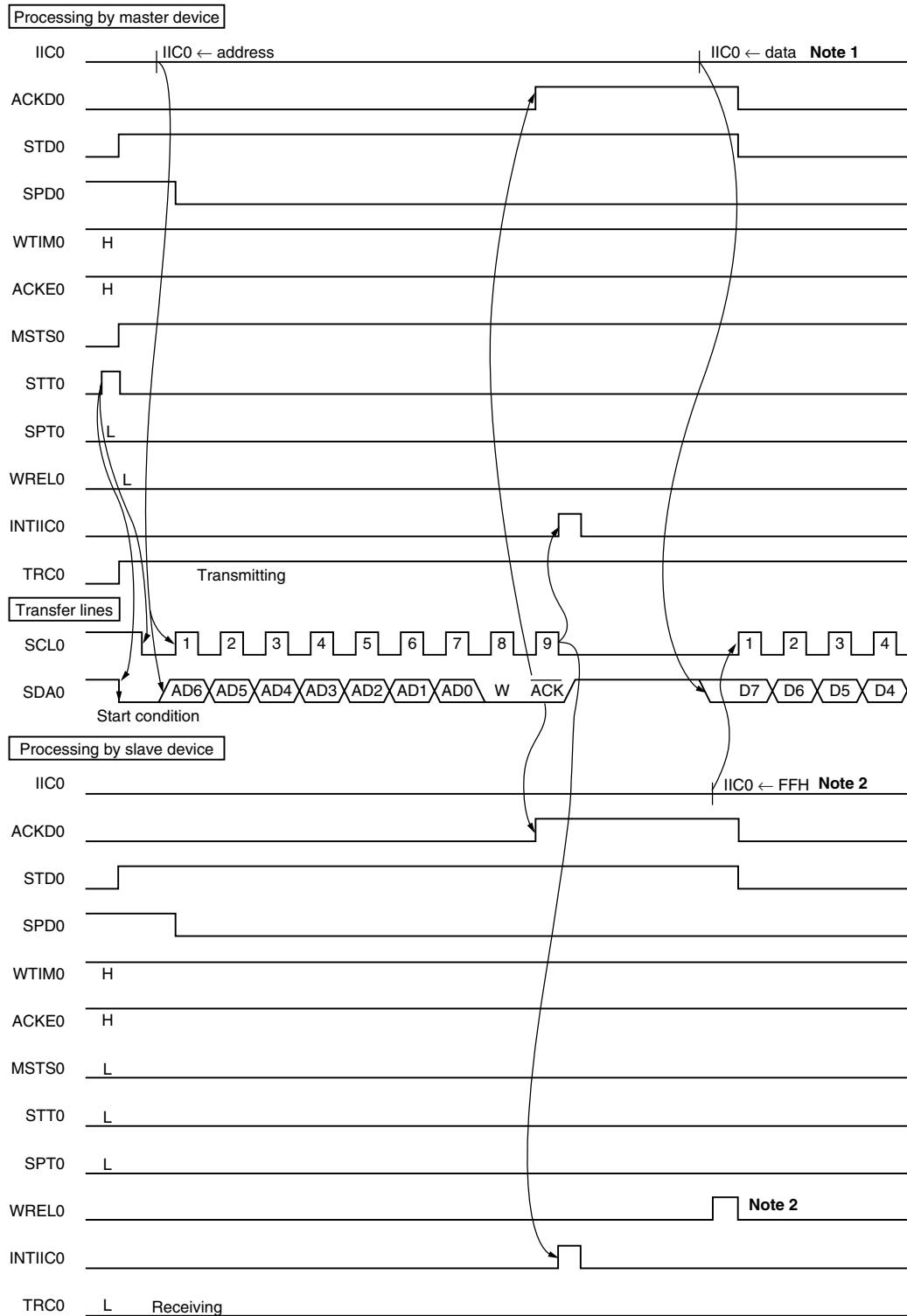
Figures 14-28 and 14-29 show timing charts of the data communication.

IIC shift register 0 (IIC0)'s shift operation is synchronized with the falling edge of the serial clock (SCL0). The transmit data is transferred to the SO0 latch and is output (MSB first) via the SDA0 pin.

Data input via the SDA0 pin is captured into IIC0 at the rising edge of SCL0.

**Figure 14-28. Example of Master to Slave Communication  
(When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)**

**(1) Start condition ~ address**

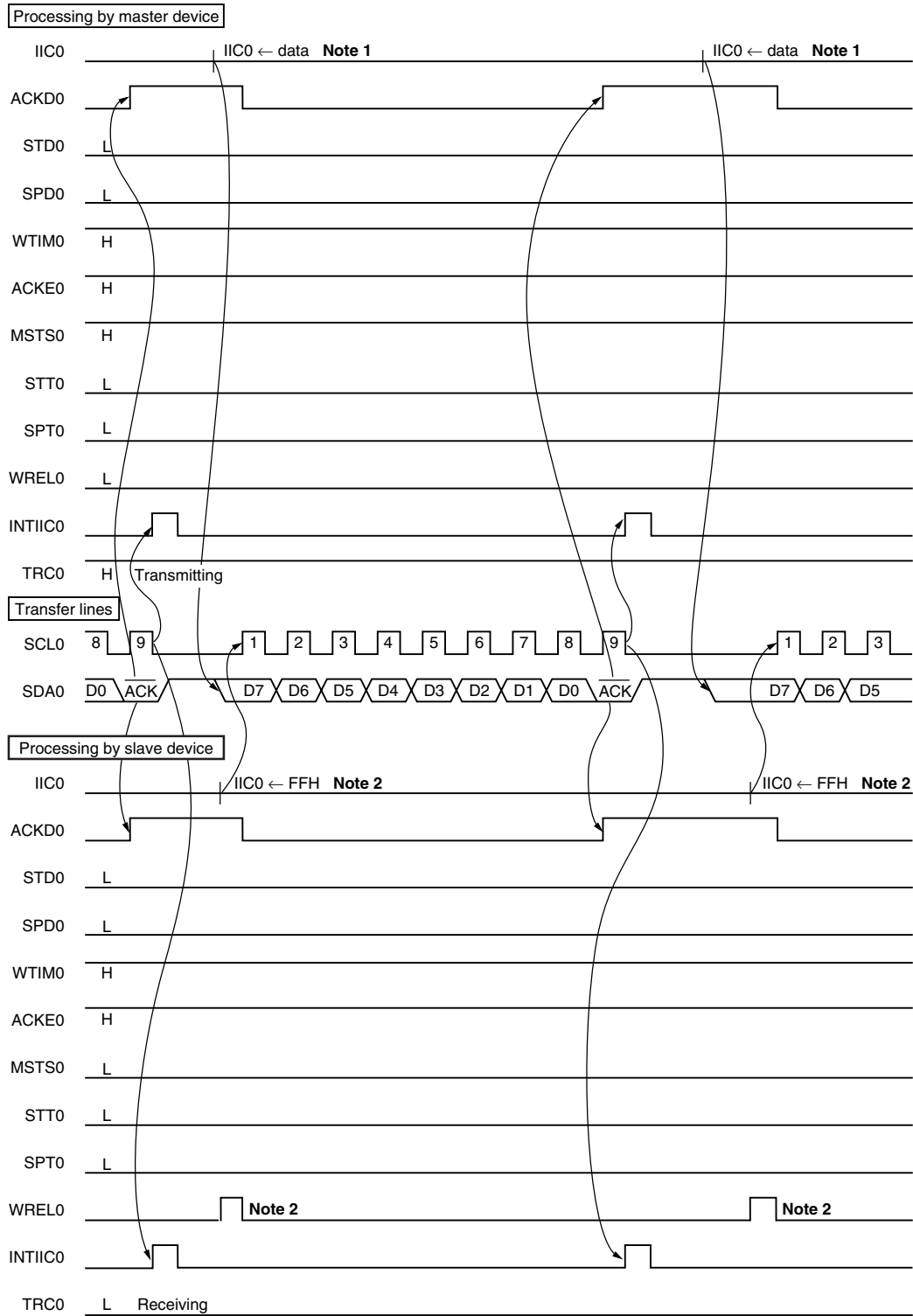


- Notes**
1. Write data to IIC0, not setting WREL0, in order to cancel a wait state during master transmission.
  2. To cancel a slave wait state, write "FFH" to IIC0 or set WREL0.



**Figure 14-28. Example of Master to Slave Communication  
(When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)**

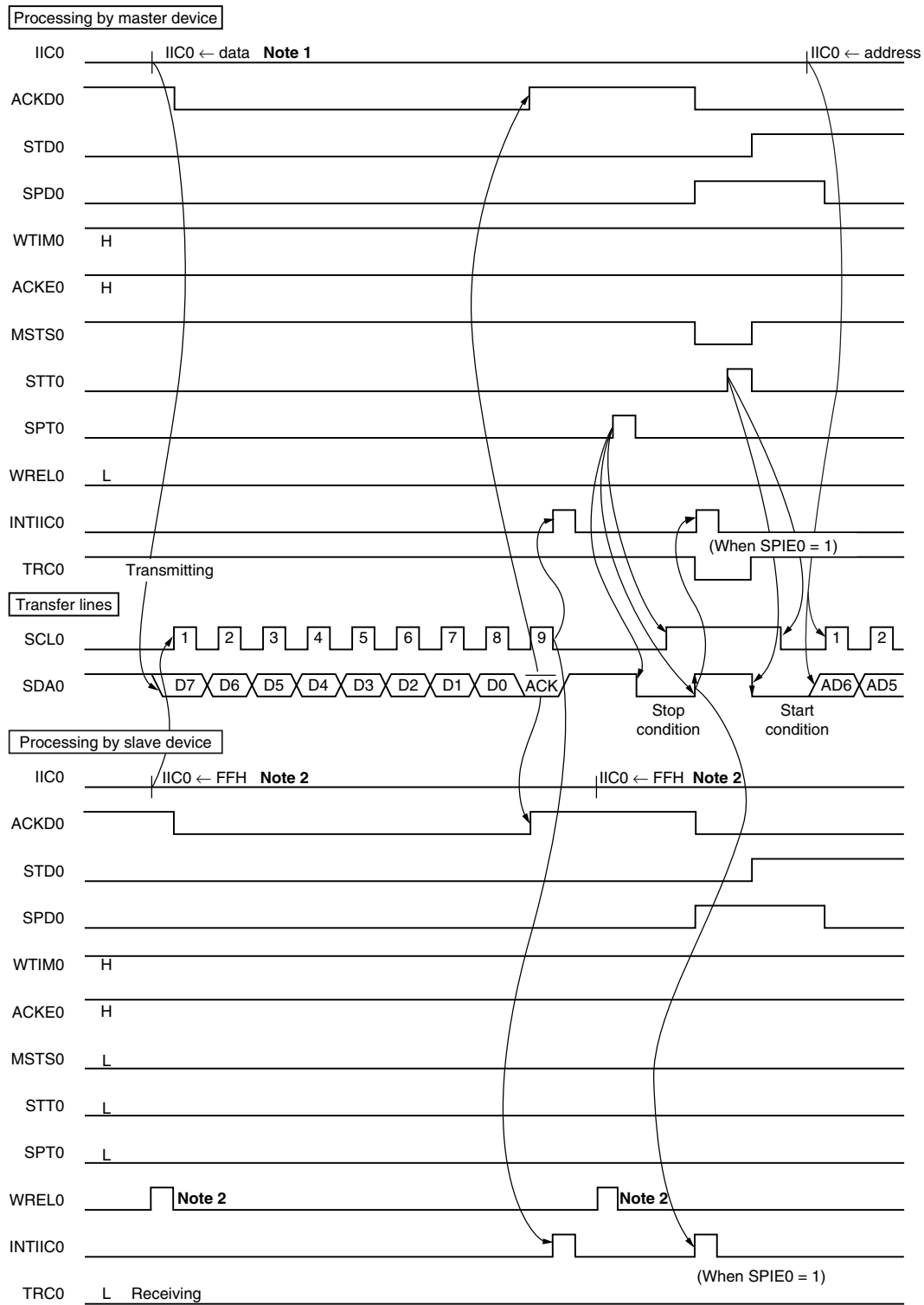
**(2) Data**



- Notes**
1. Write data to IIC0, not setting WRELO, in order to cancel a wait state during master transmission.
  2. To cancel a slave wait state, write "FFH" to IIC0 or set WRELO.

**Figure 14-28. Example of Master to Slave Communication  
(When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)**

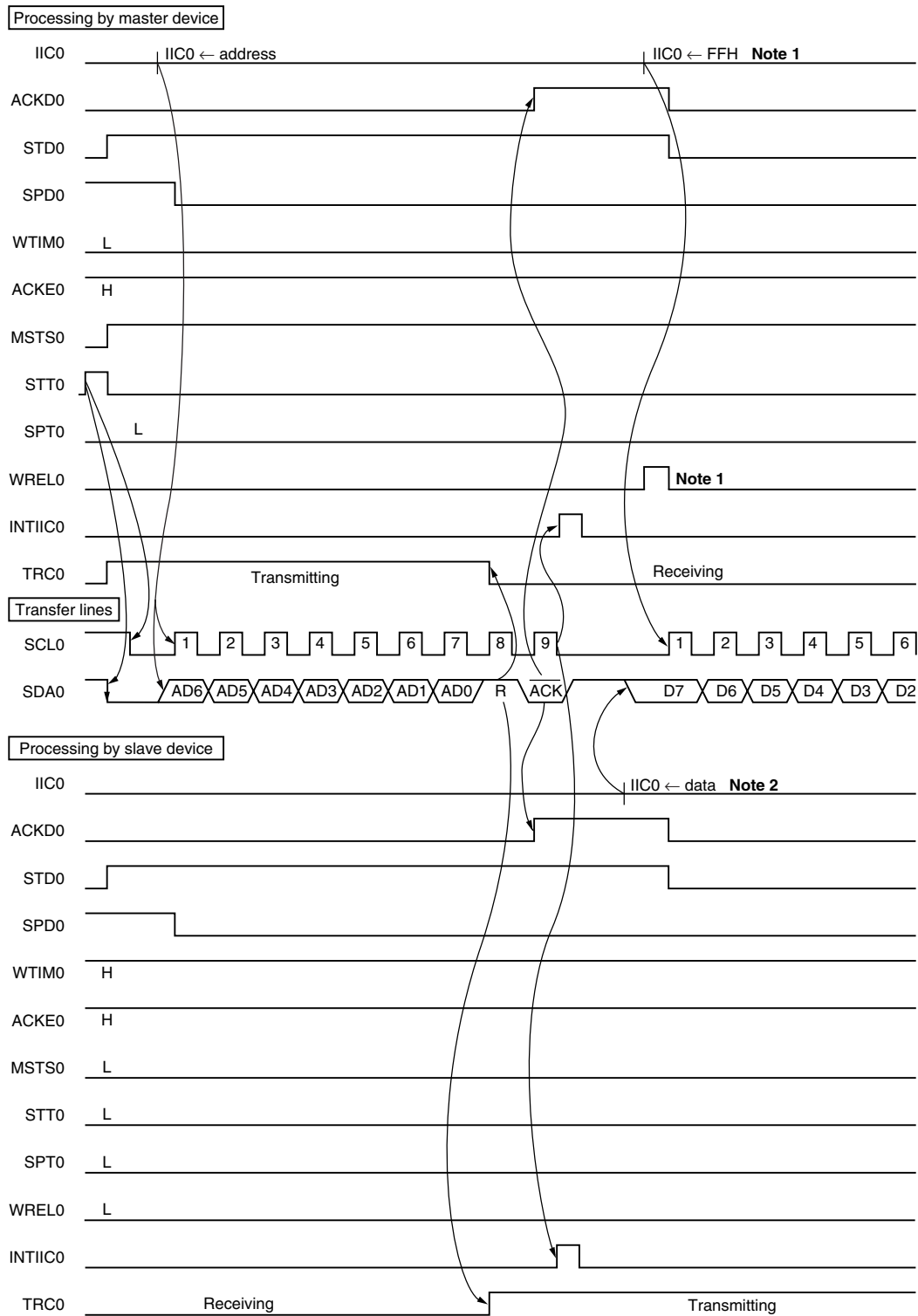
**(3) Stop condition**



- Notes**
1. Write data to IIC0, not setting WRELO, in order to cancel a wait state during master transmission.
  2. To cancel a slave wait state, write "FFH" to IIC0 or set WRELO.

**Figure 14-29. Example of Slave to Master Communication**  
**(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)**

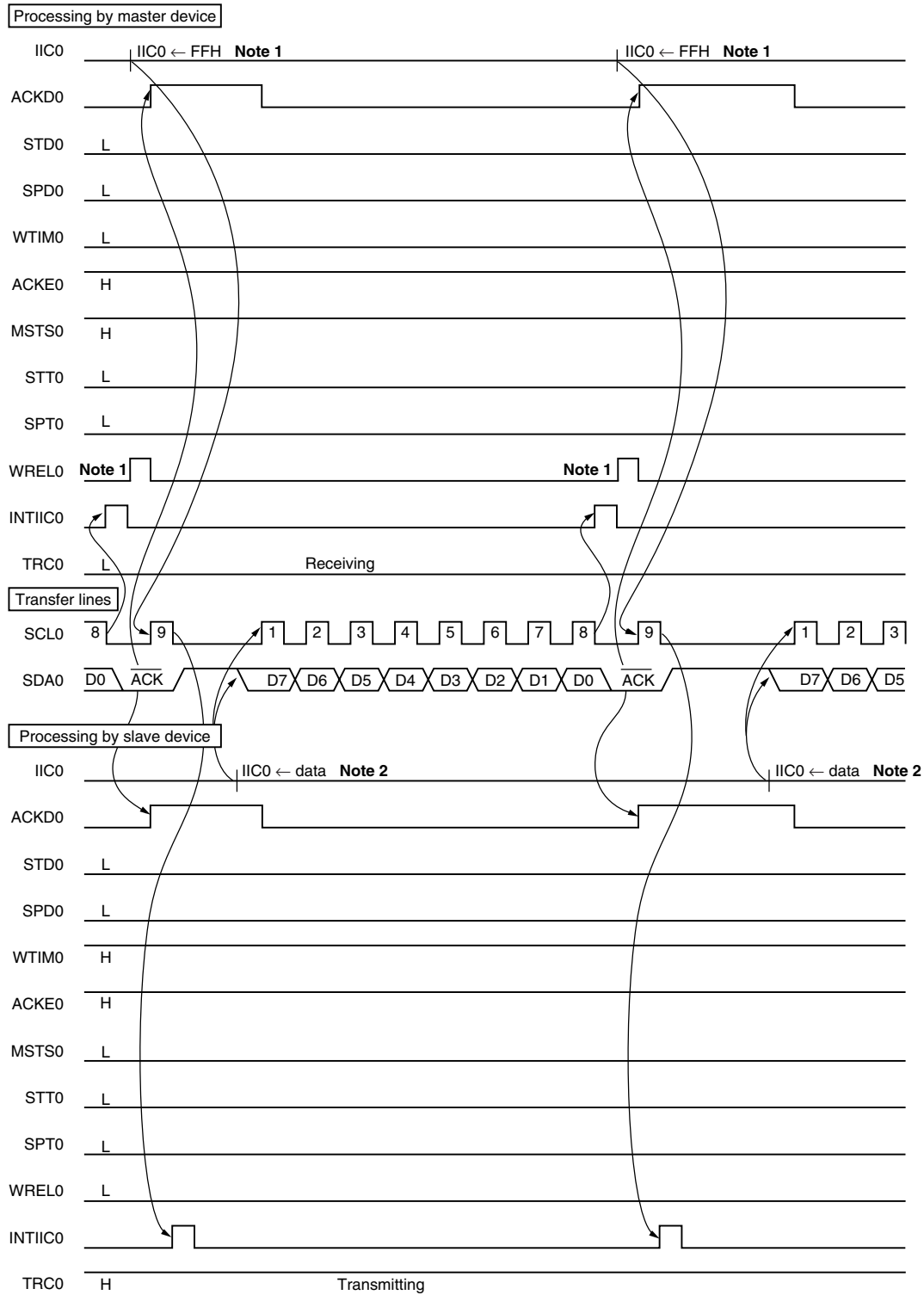
**(1) Start condition ~ address**



- Notes**
1. To cancel a master wait state, write "FFH" to IIC0 or set WREL0.
  2. Write data to IIC0, not setting WREL0, in order to cancel a wait state during slave transmission.

**Figure 14-29. Example of Slave to Master Communication**  
**(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)**

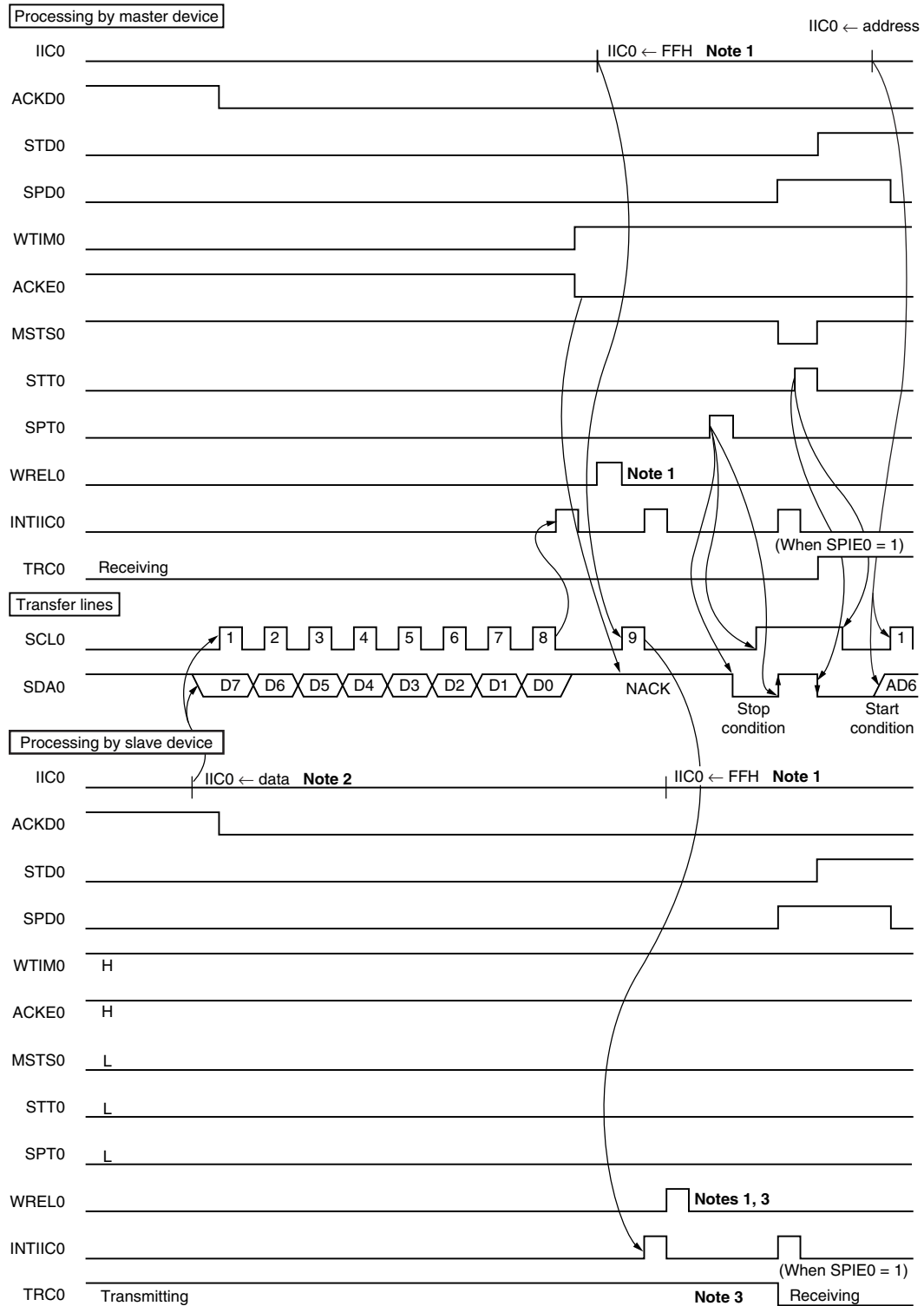
**(2) Data**



- Notes 1.** To cancel a master wait state, write “FFH” to IIC0 or set WRELO.  
**2.** Write data to IIC0, not setting WRELO, in order to cancel a wait state during slave transmission.

**Figure 14-29. Example of Slave to Master Communication**  
 (When 8-Clock and 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/3)

**(3) Stop condition**



- Notes 1.** To cancel a wait state, write "FFH" to IIC0 or set WREL0.  
**2.** Write data to IIC0, not setting WREL0, in order to cancel a wait state during slave transmission.  
**3.** If a wait state during slave transmission is canceled by setting WREL0, TRC0 will be cleared.

## CHAPTER 15 MULTIPLIER

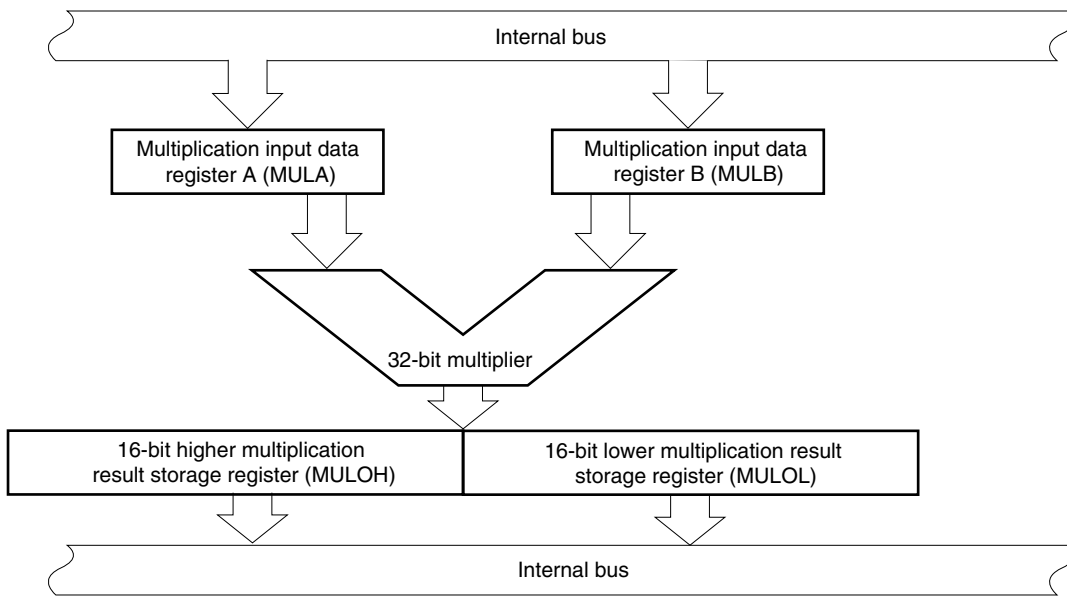
### 15.1 Functions of Multiplier

The multiplier has the following functions.

- Can execute calculation of  $16 \text{ bits} \times 16 \text{ bits} = 32 \text{ bits}$ .

Figure 15-1 shows the block diagram of the multiplier.

**Figure 15-1. Block Diagram of Multiplier**



## 15.2 Configuration of Multiplier

### (1) 16-bit higher multiplication result storage register and 16-bit lower multiplication result storage register (MULOH, MULOL)

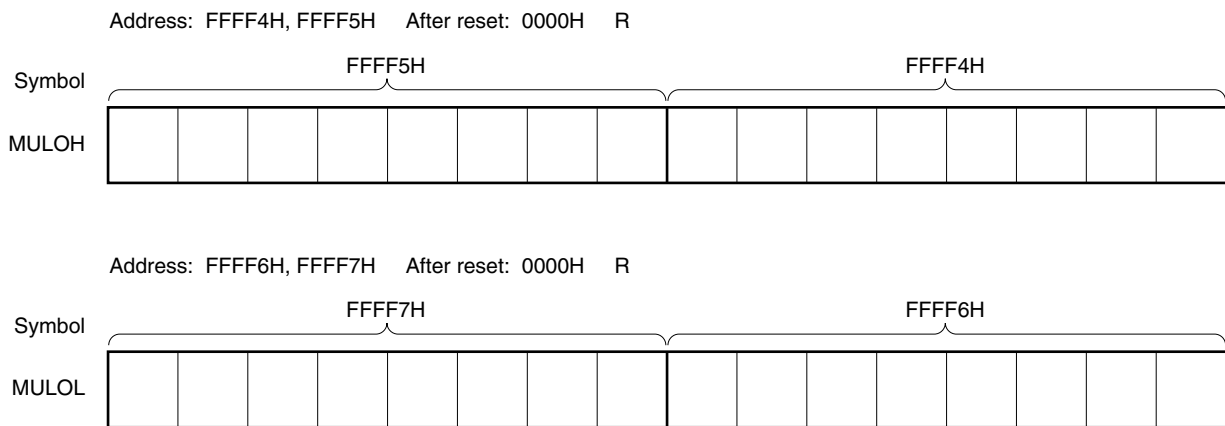
These two registers, MULOH and MULOL, are used to store a 32-bit multiplication result. The higher 16 bits of the multiplication result are stored in MULOH and the lower 16 bits, in MULOL, so that a total of 32 bits of the multiplication result can be stored.

These registers hold the result of multiplication after the lapse of one CPU clock.

MULOH and MULOL can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

**Figure 15-2. Format of 16-bit higher multiplication result storage register and 16-bit lower multiplication result storage register (MULOH, MULOL)**



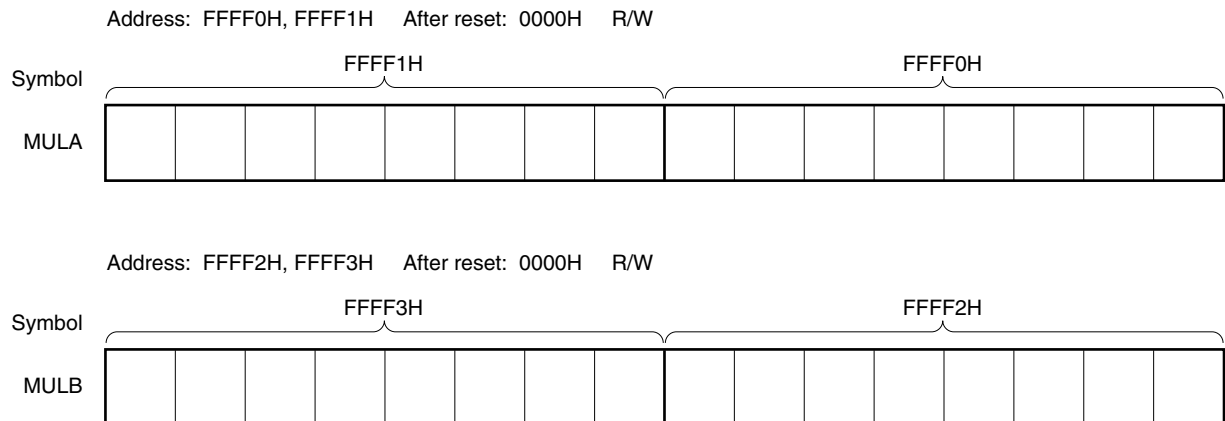
### (2) Multiplication input data registers A, B (MULA, MULB)

These are 16-bit registers that store data for multiplication. The multiplier multiplies the values of MULA and MULB.

MULA and MULB can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

**Figure 15-3. Format of Multiplication input data registers A, B (MULA, MULB)**



### 15.3 Operation of Multiplier

The result of the multiplication can be obtained by storing the values in the MULA and MULB registers and then reading the MULOH and MULOL registers after waiting for 1 clock. The result can also be obtained after 1 clock or more has elapsed, even when fixing either of MULA or MULB and rewrite the other of these. The result can be read without problem, regardless of whether MULOH or MULOL is read in first.

A source example is shown below.

Example

```
MOVW      MULA, #1234H
MOVW      MULB, #5678H
NOP                               ; 1 clock wait. Doesn't have to be NOP
MOVW      AX, MULOH                ; The result obtained on upper side
PUSH      AX
MOVW      AX, MULOL                ; The result obtained on lower side
```



## CHAPTER 16 DMA CONTROLLER

The 78K0R/KH3 has an internal DMA (Direct Memory Access) controller.

Data can be automatically transferred between the peripheral hardware supporting DMA, SFRs, and internal RAM without via CPU.

As a result, the normal internal operation of the CPU and data transfer can be executed in parallel with transfer between the SFR and internal RAM, and therefore, a large capacity of data can be processed. In addition, real-time control using communication, timer, and A/D can also be realized.

### 16.1 Functions of DMA Controller

- Number of DMA channels: 2
- Transfer unit: 8 or 16 bits
- Maximum transfer unit: 1024 times
- Transfer type: 2-cycle transfer (One transfer is processed in 2 clocks and the CPU stops during that processing.)
- Transfer mode: Single-transfer mode
- Transfer request: Selectable from the following peripheral hardware interrupts
  - A/D converter
  - Serial interface (CIS00, CSI01, CSI10, CSI11, UART0, UART1, UART3, or IIC10, IIC11)
  - Timer (channel 0, 1, 4, or 5 of timer array unit 0)
- Transfer target: Between SFR and internal RAM

Here are examples of functions using DMA.

- Successive transfer of serial interface
- Batch transfer of analog data
- Capturing A/D conversion result at fixed interval
- Capturing port value at fixed interval

## 16.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

**Table 16-1. Configuration of DMA Controller**

Item	Configuration
Address registers	<ul style="list-style-type: none"> <li>• DMA SFR address registers 0, 1 (DSA0, DSA1)</li> <li>• DMA RAM address registers 0, 1 (DRA0, DRA1)</li> </ul>
Count register	<ul style="list-style-type: none"> <li>• DMA byte count registers 0, 1 (DBC0, DBC1)</li> </ul>
Control registers	<ul style="list-style-type: none"> <li>• DMA mode control registers 0, 1 (DMC0, DMC1)</li> <li>• DMA operation control register 0, 1 (DRC0, DRC1)</li> </ul>

### (1) DMA SFR address register n (DSAn)

This is an 8-bit register that is used to set an SFR address that is the transfer source or destination of DMA channel n.

Set the lower 8 bits of the SFR addresses FFF00H to FFFFFH<sup>Note</sup>.

This register is not automatically incremented but fixed to a specific value.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

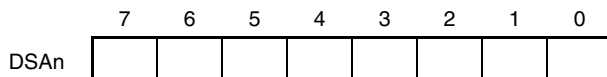
DSAn can be read or written in 8-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 00H.

**Note** Except for address FFFFEH because the PMC register is allocated there.

**Figure 16-1. Format of DMA SFR Address Register n (DSAn)**

Address: FFFB0H (DSA0), FFFB1H (DSA1) After reset: 00H R/W



**Remark** n: DMA channel number (n = 0, 1)

**(2) DMA RAM address register n (DRAn)**

This is a 16-bit register that is used to set a RAM address that is the transfer source or destination of DMA channel n.

Addresses of the internal RAM area other than the general-purpose registers (FDF00H to FFEDFH in the case of the  $\mu$ PD78F1174, 78F1174A) can be set to this register.

Set the lower 16 bits of the RAM address.

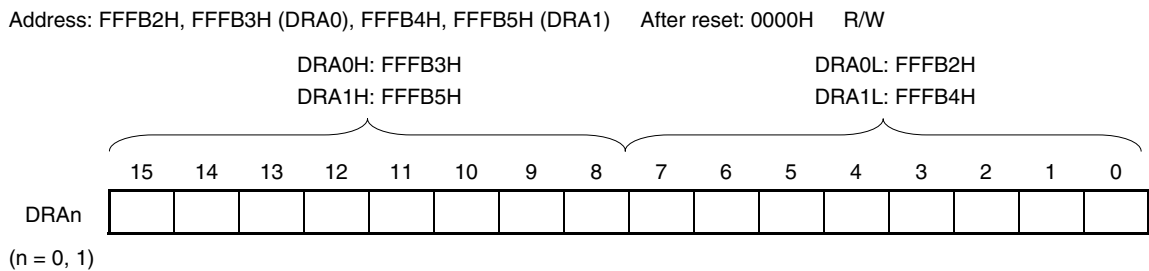
This register is automatically incremented when DMA transfer has been started. It is incremented by +1 in the 8-bit transfer mode and by +2 in the 16-bit transfer mode. DMA transfer is started from the address set to this DRAn register. When the data of the last address has been transferred, DRAn stops with the value of the last address +1 in the 8-bit transfer mode, and the last address +2 in the 16-bit transfer mode.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

DRAn can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

**Figure 16-2. Format of DMA RAM Address Register n (DRAn)**

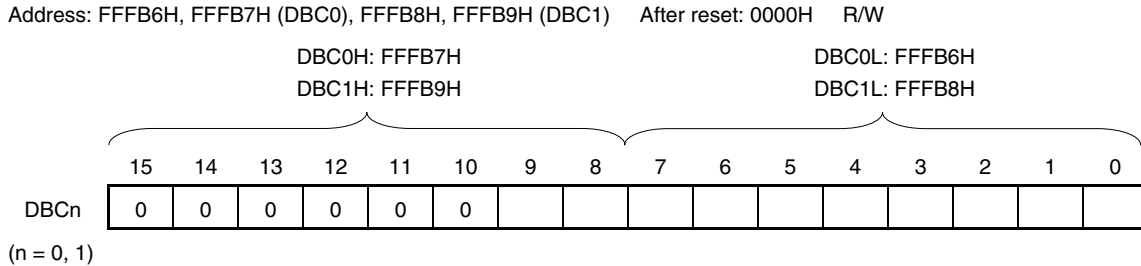


**Remark**    n: DMA channel number (n = 0, 1)

**(3) DMA byte count register n (DBCn)**

This is a 10-bit register that is used to set the number of times DMA channel n executes transfer. Be sure to set the number of times of transfer to this DBCn register before executing DMA transfer (up to 1024 times). Each time DMA transfer has been executed, this register is automatically decremented. By reading this DBCn register during DMA transfer, the remaining number of times of transfer can be learned. DBCn can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer. Reset signal generation clears this register to 0000H.

**Figure 16-3. Format of DMA Byte Count Register n (DBCn)**



DBCn[9:0]	Number of Times of Transfer (When DBCn is Written)	Remaining Number of Times of Transfer (When DBCn is Read)
000H	1024	Completion of transfer or waiting for 1024 times of DMA transfer
001H	1	Waiting for remaining one time of DMA transfer
002H	2	Waiting for remaining two times of DMA transfer
003H	3	Waiting for remaining three times of DMA transfer
•	•	•
•	•	•
•	•	•
3FEH	1022	Waiting for remaining 1022 times of DMA transfer
3FFH	1023	Waiting for remaining 1023 times of DMA transfer

- Cautions**
1. Be sure to clear bits 15 to 10 to “0”.
  2. If the general-purpose register is specified or the internal RAM space is exceeded as a result of continuous transfer, the general-purpose register or SFR space are written or read, resulting in loss of data in these spaces. Be sure to set the number of times of transfer that is within the internal RAM space.

**Remark**    n: DMA channel number (n = 0, 1)

### 16.3 Registers to Controlling DMA Controller

DMA controller is controlled by the following registers.

- DMA mode control register n (DMCn)
- DMA operation control register n (DRCn)

**Remark** n: DMA channel number (n = 0, 1)

#### (1) DMA mode control register n (DMCn)

DMCn is a register that is used to set a transfer mode of DMA channel n. It is used to select a transfer direction, data size, setting of pending, and start source. Bit 7 (STGn) is a software trigger that starts DMA.

Rewriting bits 6, 5, and 3 to 0 of DMCn is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

**Figure 16-4. Format of DMA Mode Control Register n (DMCn) (1/2)**

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

STGn <sup>Note 1</sup>	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).
DMA transfer is performed once by writing 1 to STGn when DMA operation is enabled (DENn = 1). When this bit is read, 0 is always read.	

DRSn	Selection of DMA transfer direction
0	SFR to internal RAM
1	Internal RAM to SFR

DSn	Specification of transfer data size for DMA transfer
0	8 bits
1	16 bits

DWAITn <sup>Note 2</sup>	Pending of DMA transfer
0	Executes DMA transfer upon DMA start request (not held pending).
1	Holds DMA start request pending if any.
DMA transfer that has been held pending can be started by clearing the value of DWAITn to 0. It takes 2 clocks to actually hold DMA transfer pending when the value of DWAITn is set to 1.	

**Notes 1.** The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 values.

**2.** When DMA transfer is held pending while using both DMA channels, be sure to hold the DMA transfer pending for both channels (by setting DWAIT0 and DWAIT1 to 1).

**Remark** n: DMA channel number (n = 0, 1)

<R>

Figure 16-4. Format of DMA Mode Control Register n (DMCn) (2/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

IFCn 3	IFCn 2	IFCn 1	IFCn 0	Selection of DMA stat source <sup>Note</sup>	
				Trigger signal	Trigger contents
0	0	0	0	–	Disables DMA transfer by interrupt. (Only software trigger is enabled.)
0	0	1	0	INTTM00	End of timer channel 0 count or capture end interrupt
0	0	1	1	INTTM01	End of timer channel 1 count or capture end interrupt
0	1	0	0	INTTM04	End of timer channel 4 count or capture end interrupt
0	1	0	1	INTTM05	End of timer channel 5 count or capture end interrupt
0	1	1	0	INTST0/INTCSI00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt
0	1	1	1	INTSR0/INTCSI01	UART0 reception transfer end interrupt/CSI01 transfer end or buffer empty interrupt
1	0	0	0	INTST1/INTCSI10/INTIIC10	UART1 transmission transfer end or buffer empty interrupt/CSI10 transfer end or buffer empty interrupt/ IIC10 transfer end interrupt
1	0	0	1	INTSR1/INTCSI11/INTIIC11	UART1 reception transfer end interrupt/CSI11 transfer end or buffer empty interrupt/ IIC11 transfer end interrupt
1	0	1	0	INTST3	UART3 transmission transfer end or buffer empty interrupt
1	0	1	1	INTSR3	UART3 reception transfer end interrupt
1	1	0	0	INTAD	A/D conversion end interrupt
Other than above				Setting prohibited	

**Note** The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 values.

**Remark** n: DMA channel number (n = 0, 1)

**(2) DMA operation control register n (DRCn)**

DRCn is a register that is used to enable or disable transfer of DMA channel n.  
 Rewriting bit 7 (DENn) of this register is prohibited during operation (when DSTn = 1).  
 DRCn can be set by a 1-bit or 8-bit memory manipulation instruction.  
 Reset signal generation clears this register to 00H.

**Figure 16-5. Format of DMA Operation Control Register n (DRCn)**

Address: FFFBCH (DRC0), FFFBDH (DRC1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA operation enable flag
0	Disables operation of DMA channel n (stops operating cock of DMA).
1	Enables operation of DMA channel n.
The DMA controller waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1).	

DSTn	DMA transfer mode flag
0	DMA transfer of DMA channel n is completed.
1	DMA transfer of DMA channel n is not completed (still under execution).
The DMA controller waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1). When a software trigger (STGn) or the start source trigger set by IFCn3 to IFCn0 is input, DMA transfer is started. When DMA transfer is completed after that, this bit is automatically cleared to 0. Write 0 to this bit to forcibly terminate DMA transfer under execution.	

**Caution** The DSTn flag is automatically cleared to 0 when a DMA transfer is completed.  
 Writing the DENn flag is enabled only when DSTn = 0. When a DMA transfer is terminated without waiting for generation of the interrupt (INTDMA<sub>n</sub>) of DMA<sub>n</sub>, therefore, set DSTn to 0 and then DENn to 0 (for details, refer to 16.5.7 Forced termination by software).

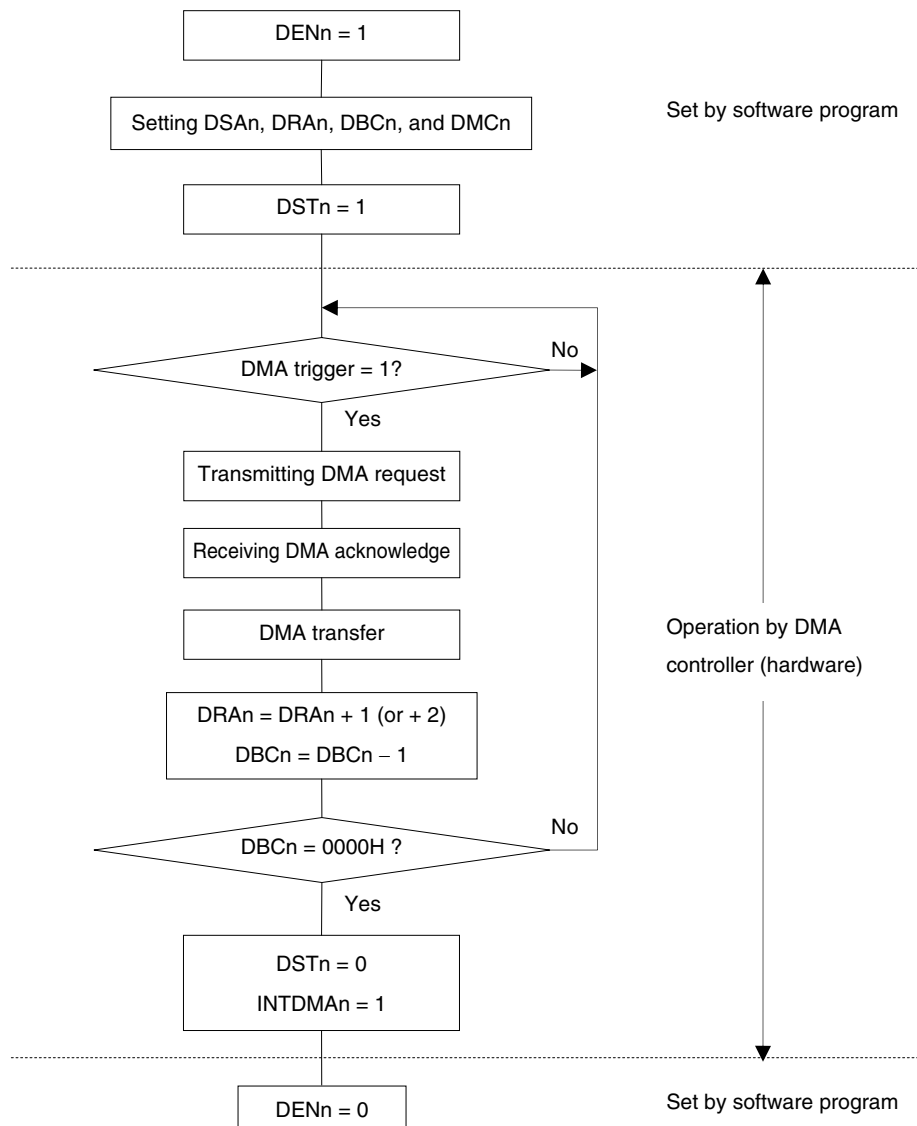
**Remark** n: DMA channel number (n = 0, 1)

## 16.4 Operation of DMA Controller

### 16.4.1 Operation procedure

- <1> The DMA controller is enabled to operate when  $DEN_n = 1$ . Before writing the other registers, be sure to set  $DEN_n$  to 1. Use 80H to write with an 8-bit manipulation instruction.
- <2> Set an SFR address, a RAM address, the number of times of transfer, and a transfer mode of DMA transfer to the  $DSAn$ ,  $DRAn$ ,  $DBCn$ , and  $DMCn$  registers.
- <3> The DMA controller waits for a DMA trigger when  $DSTn = 1$ . Use 81H to write with an 8-bit manipulation instruction.
- <4> When a software trigger ( $STGn$ ) or a start source trigger specified by  $IFCn3$  to  $IFCn0$  is input, a DMA transfer is started.
- <5> Transfer is completed when the number of times of transfer set by the  $DBCn$  register reaches 0, and transfer is automatically terminated by occurrence of an interrupt ( $INTDMA_n$ ).
- <6> Stop the operation of the DMA controller by clearing  $DEN_n$  to 0 when the DMA controller is not used.

**Figure 16-6. Operation Procedure**



**Remark** n: DMA channel number (n = 0, 1)



### 16.4.2 Transfer mode

The following four modes can be selected for DMA transfer by using bits 6 and 5 (DRSn and DS<sub>n</sub>) of the DMC<sub>n</sub> register.

DRS <sub>n</sub>	DS <sub>n</sub>	DMA Transfer Mode
0	0	Transfer from SFR of 1-byte data (fixed address) to RAM (address is incremented by +1)
0	1	Transfer from SFR of 2-byte data (fixed address) to RAM (address is incremented by +2)
1	0	Transfer from RAM of 1-byte data (address is incremented by +1) to SFR (fixed address)
1	1	Transfer from RAM of 2-byte data (address is incremented by +2) to SFR (fixed address)

By using these transfer modes, up to 1024 bytes of data can be consecutively transferred by using the serial interface, data resulting from A/D conversion can be consecutively transferred, and port data can be scanned at fixed time intervals by using a timer.

### 16.4.3 Termination of DMA transfer

When DBC<sub>n</sub> = 00H and DMA transfer is completed, the DST<sub>n</sub> bit is automatically cleared to 0. An interrupt request (INTDMA<sub>n</sub>) is generated and transfer is terminated.

When the DST<sub>n</sub> bit is cleared to 0 to forcibly terminate DMA transfer, the DBC<sub>n</sub> and DRAn registers hold the value when transfer is terminated.

The interrupt request (INTDMA<sub>n</sub>) is not generated if transfer is forcibly terminated.

**Remark** n: DMA channel number (n = 0, 1)

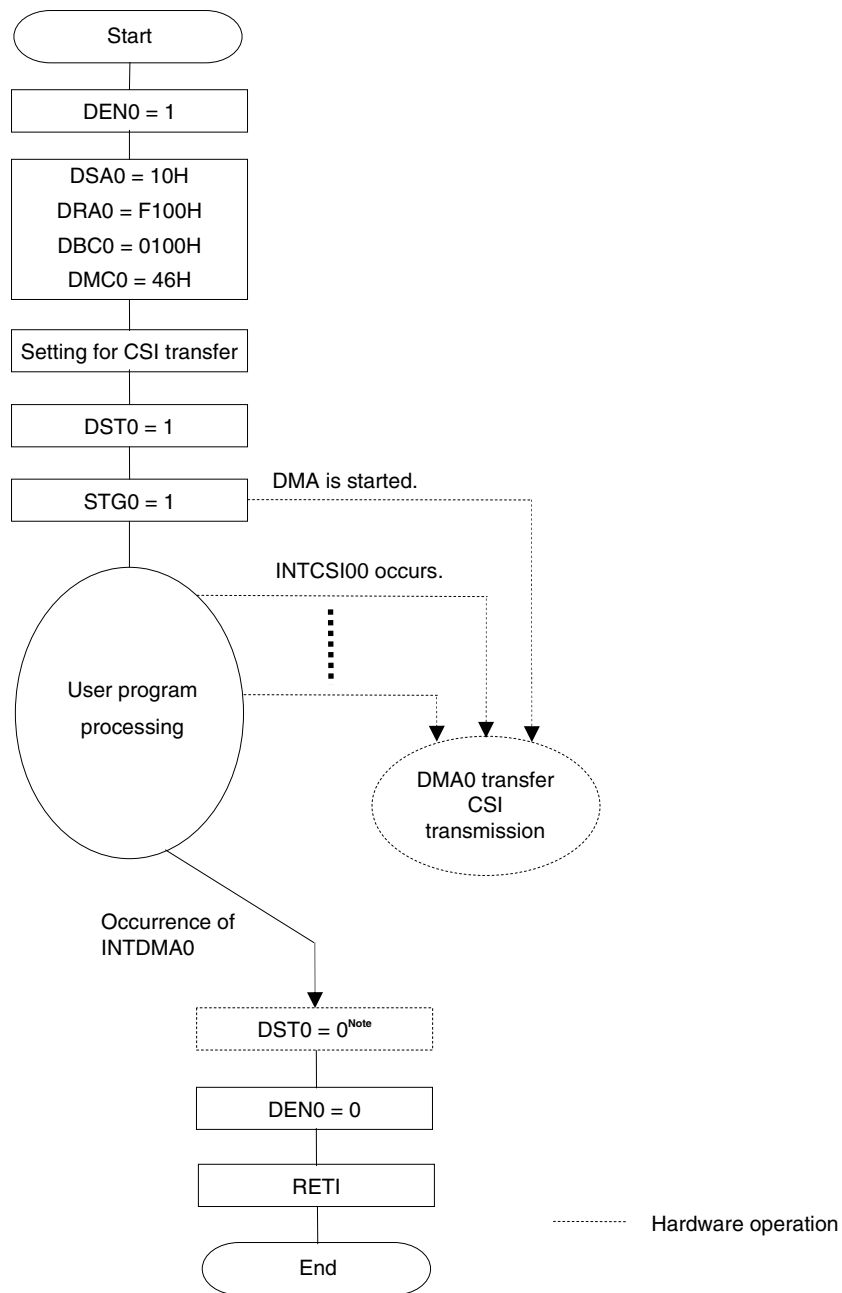
## 16.5 Example of Setting of DMA Controller

### 16.5.1 CSI consecutive transmission

A flowchart showing an example of setting for CSI consecutive transmission is shown below.

- Consecutive transmission (256 bytes) of CSI00
  - DMA channel 0 is used for DMA transfer.
  - DMA start source: INTCSI00 (software trigger (STG0) only for the first start source)
  - Interrupt of CSI00 is specified by IFC03 to IFC00 (bits 3 to 0 of the DMC0 register) = 0110B.
- <R> • Transfers FF100H to FF1FFH (256 bytes) of RAM to FFF10H of the data register (SIO00) of CSI.

Figure 16-7. Setting Example of CSI Consecutive Transmission



**Note** The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.

Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set DST0 to 0 and then DEN0 to 0 (for details, refer to **16.5.7 Forcible termination by software**).

<R> The first trigger for consecutive transmission is not started by the interrupt of CSI. In this example, it starts by a software trigger.

CSI transmission of the second time and onward is automatically executed.

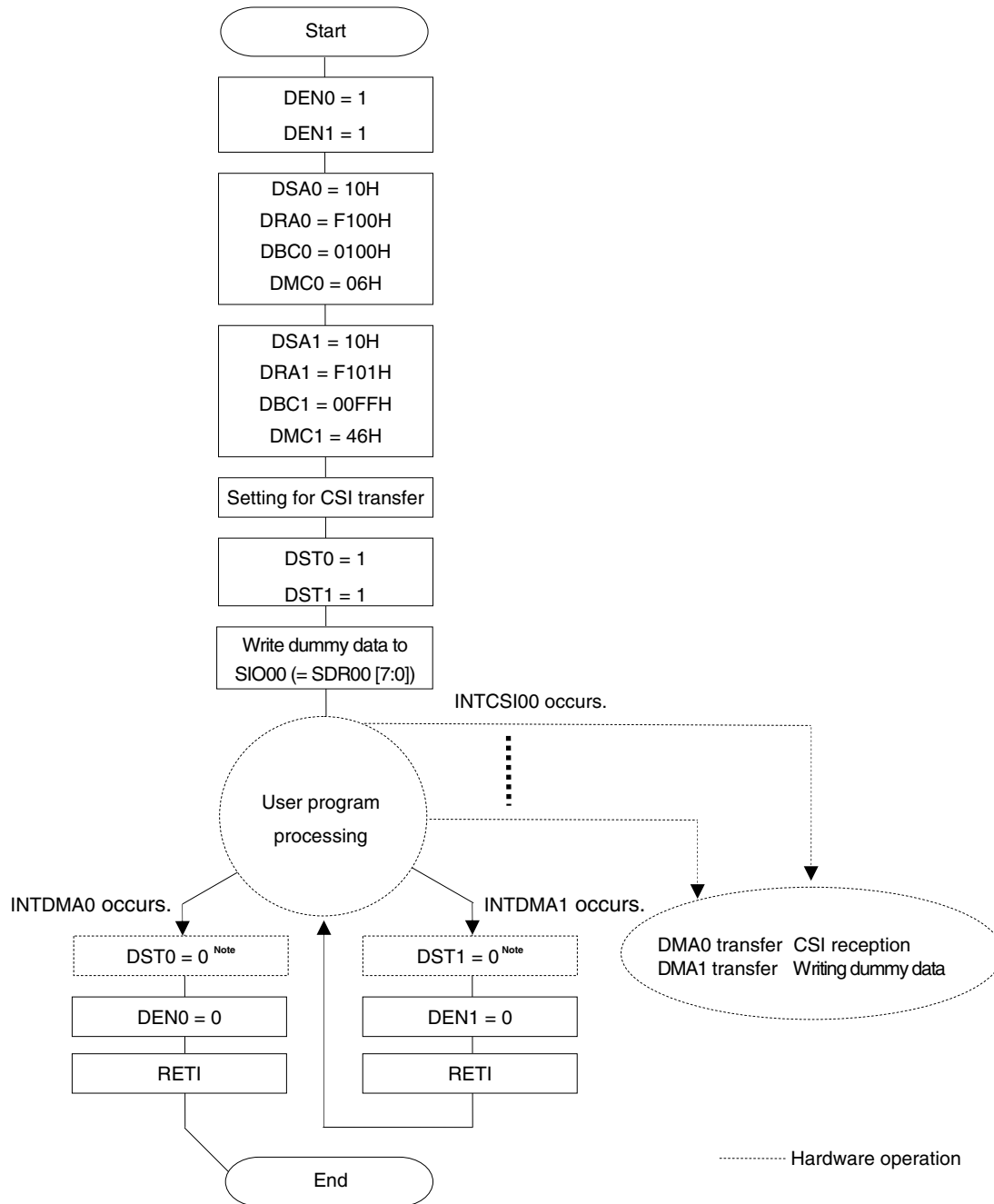
A DMA interrupt (INTDMA0) occurs when the last transmit data has been written to the data register.

**<R> 16.5.2 CSI master reception**

A flowchart showing an example of setting for CSI master reception is shown below.

- Master reception (256 bytes) of CSI00
- DMA channel 0 is used to read received data and DMA channel 1 is used to write dummy data.
- DMA start source: INTCSI00  
(If the same start source is specified for DMA channels 0 and 1, the data of channel 0 is transferred, and then that of channel 1.)
- Interrupt of CSI00 is specified by IFC03 to IFC00 = IFC13 to IFC10 (bits 3 to 0 of the DMCn register) = 0110B.
- Data is transferred (received) from FFF10H of the CSI data register (SIO00) to FF100H to FF1FFH of RAM (256 bytes). (In successive reception mode, the data that is to be received when the first buffer empty interrupt occurs is invalid because the data has not been received.)
- Transfers dummy data FF101H to FF1FFH (255 bytes) of RAM to FFF10H of the data register (SIO00) of CSI. (Dummy data is written to the first byte by using software (an instruction).)

Figure 16-8. Setting Example of CSI Master Reception



**Note** The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENn flag is enabled only when DSTn = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA<sub>n</sub> (INTDMA<sub>n</sub>), set DSTn to 0 and then DENn to 0 (for details, refer to **16.5.7 Forcible termination by software**).

Because no CSI interrupt is generated when reception starts during CSI master reception, dummy data is written using software in this example.

The received data is automatically transferred from the first byte (In successive reception mode, the data that is to be received when the first buffer empty interrupt occurs is invalid because the valid data has not been received.).

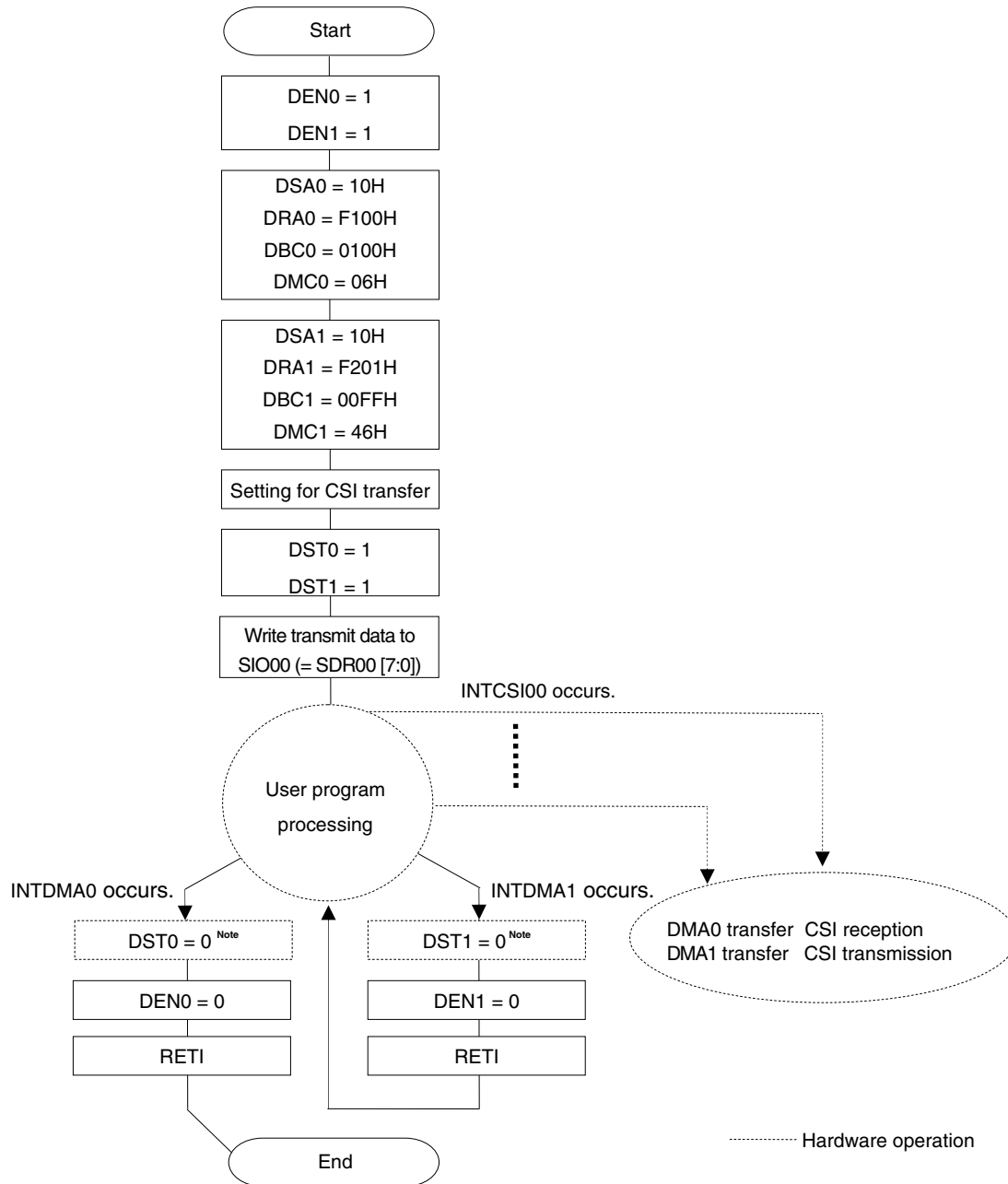
A DMA interrupt (INTDMA1) occurs when the last dummy data has been writing to the data register. A DMA interrupt (INTDMA0) occurs when the last received data has been read from the data register. To restart the DMA transfer, the CSI transfer must be completed.

**<R> 16.5.3 CSI transmission/reception**

A flowchart showing an example of setting for CSI transmission/reception is shown below.

- Transmission/reception (256 bytes) of CSI00
- DMA channel 0 is used to read received data and DMA channel 1 is used to write transmit data.
- DMA start source: INTCSI00  
(If the same start source is specified for DMA channels 0 and 1, the data of channel 0 is transferred, and then that of channel 1)
- Interrupt of CSI00 is specified by IFC03 to IFC00 = IFC13 to IFC10 (bits 3 to 0 of the DMCn register) = 0110B.
- Data is transferred (received) from FFF10H of the CSI data register (SIO00) to FF100H to FF1FFH of RAM (256 bytes). (In successive transmission/reception mode, the data that is to be received when the first buffer empty interrupt occurs is invalid because the data has not been received.)
- Transfers FF201H to FF2FFH (255 bytes) of RAM to FFF10H of the data register (SIO00) of CSI (transmission) (Transmit data is written to the first byte by using software (an instruction).)

Figure 16-9. Setting Example of CSI Transmission/reception



**Note** The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENn flag is enabled only when DSTn = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA<sub>n</sub> (INTDMA<sub>n</sub>), set DSTn to 0 and then DENn to 0 (for details, refer to 16.5.7 Forcible termination by software).

During CSI transfers, no CSI interrupt is generated when the transmitted data of the first byte is written. Therefore, the transmitted data is written using software in this example. The data of the second and following bytes is automatically transmitted.

The received data is automatically transferred from the first byte. (In successive transmission/reception, the data that is to be received when the first buffer empty interrupt occurs is invalid because the valid data has not been received.)

A DMA interrupt (INTDMA1) occurs when the last transmit data has been writing to the data register. A DMA interrupt (INTDMA0) occurs when the last received data has been read from the data register. To restart the DMA transfer, the CSI transfer must be completed.

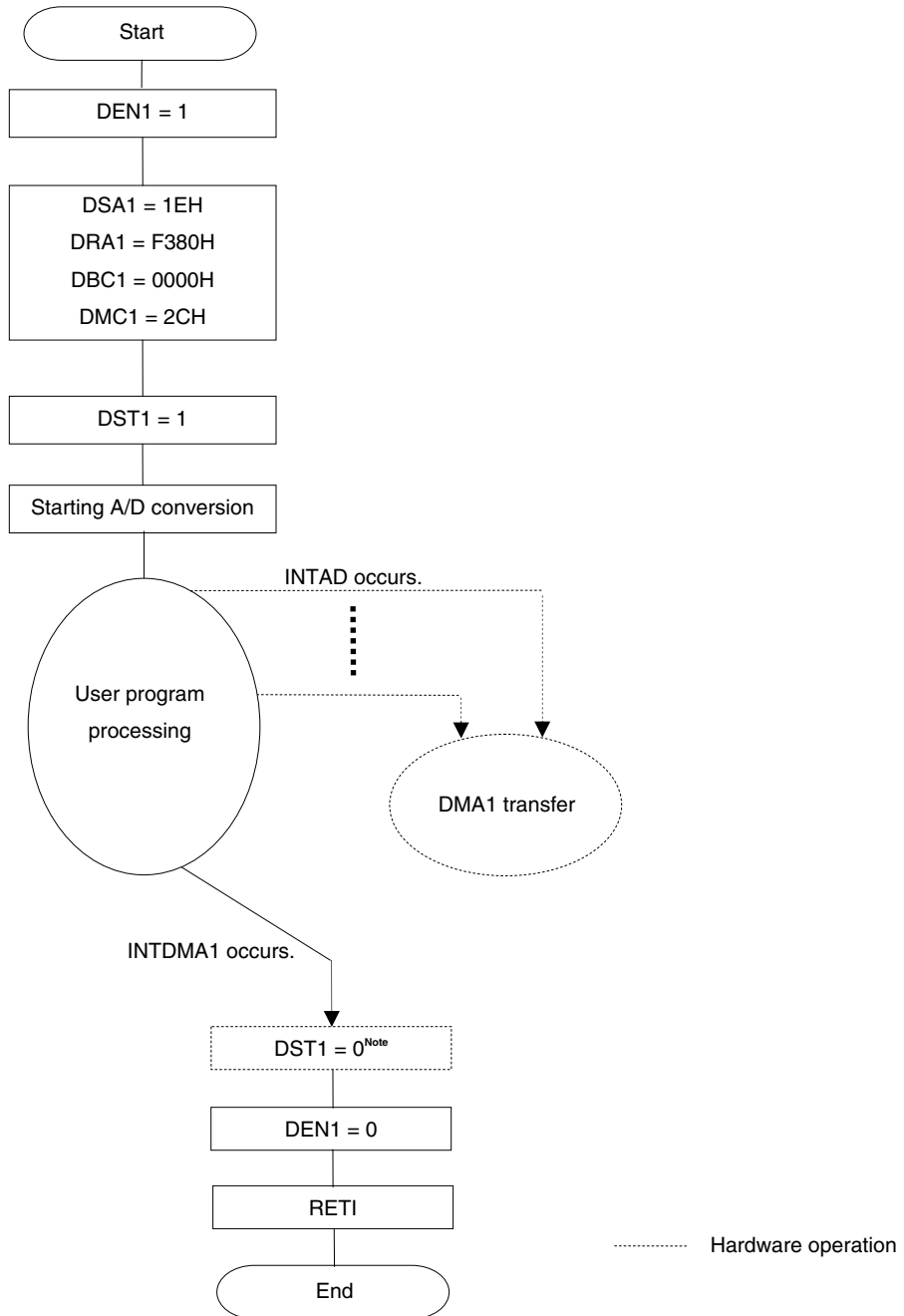
#### 16.5.4 Consecutive capturing of A/D conversion results

A flowchart of an example of setting for consecutively capturing A/D conversion results is shown below.

- Consecutive capturing of A/D conversion results.
- DMA channel 1 is used for DMA transfer.
- DMA start source: INTAD
- Interrupt of A/D is specified by IFC13 to IFC10 (bits 3 to 0 of the DMC1 register) = 1100B.
- Transfers FFF1EH and FFF1FH (2 bytes) of the 10-bit A/D conversion result register to 2048 bytes of FF380H to FFB7FH of RAM.



Figure 16-10. Setting Example of Consecutively Capturing A/D Conversion Results



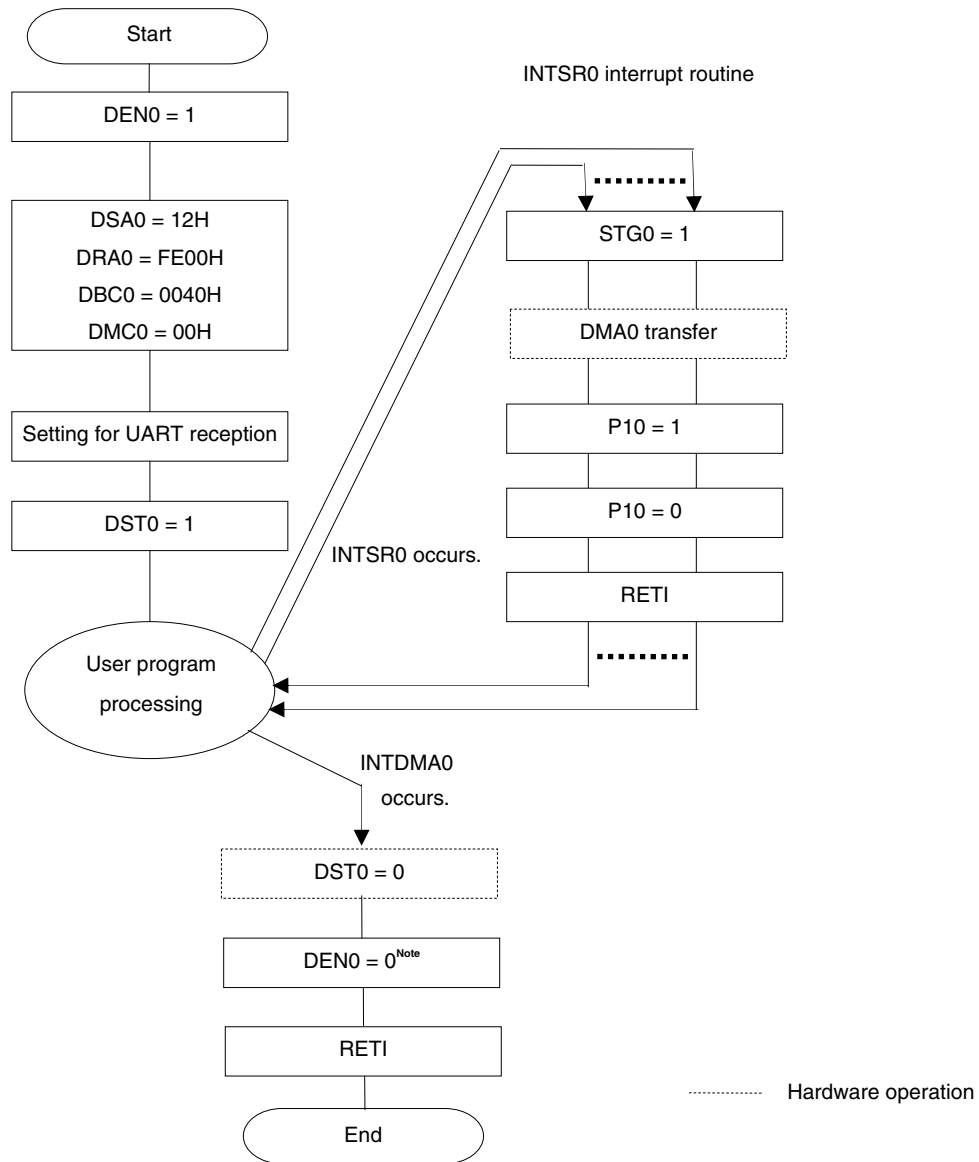
**Note** The DST1 flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set DST1 to 0 and then DEN1 to 0 (for details, refer to 16.5.7 **Forcible termination by software**).

### 16.5.5 UART consecutive reception + ACK transmission

A flowchart illustrating an example of setting for UART consecutive reception + ACK transmission is shown below.

- Consecutively receives data from UART0 and outputs ACK to P10 on completion of reception.
- DMA channel 0 is used for DMA transfer.
- DMA start source: Software trigger (DMA transfer on occurrence of an interrupt is disabled.)
- Transfers FFF12H of UART receive data register 0 (RXD0) to 64 bytes of FFE00H to FFE3FH of RAM.

Figure 16-11. Setting Example of UART Consecutive Reception + ACK Transmission



**Note** The DST0 flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENO flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set DST0 to 0 and then DENO to 0 (for details, refer to **16.5.7 Forcible termination by software**).

**Remark** This is an example where a software trigger is used as a DMA start source. If ACK is not transmitted and if only data is consecutively received from UART, the UART reception end interrupt (INTSR0) can be used to start DMA for data reception.

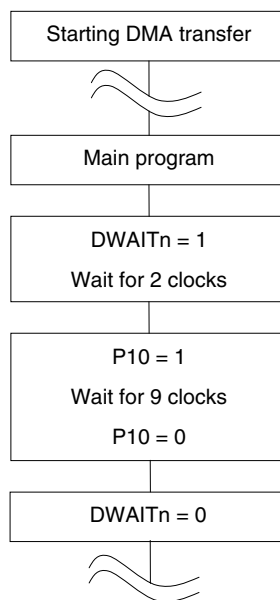
### 16.5.6 Holding DMA transfer pending by DWAITn

When DMA transfer is started, transfer is performed while an instruction is executed. At this time, the operation of the CPU is stopped and delayed for the duration of 2 clocks. If this poses a problem to the operation of the set <R> system, a DMA transfer can be held pending by setting DWAITn to 1. The DMA transfer for a transfer trigger that occurred while DMA transfer was held pending is executed after the pending status is canceled. However, because only one transfer trigger can be held pending for each channel, even if multiple transfer triggers occur for one channel during the pending status, only one DMA transfer is executed after the pending status is canceled.

To output a pulse with a width of 10 clocks of the operating frequency from the P00 pin, for example, the clock width increases to 12 if a DMA transfer is started midway. In this case, the DMA transfer can be held pending by setting DWAITn to 1.

After setting DWAITn to 1, it takes two clocks until a DMA transfer is held pending.

**Figure 16-12. Example of Setting for Holding DMA Transfer Pending by DWAITn**



<R> **Caution** When DMA transfer is held pending while using both DMA channels, be sure to hold the DMA transfer pending for both channels (by setting DWAIT0 and DWAIT1 to 1). If the DMA transfer of one channel is executed while that of the other channel is held pending, DMA transfer might not be held pending for the latter channel.

- Remarks**
1. n: DMA channel number (n = 0, 1)
  2. 1 clock:  $1/f_{CLK}$  ( $f_{CLK}$ : CPU clock)

### 16.5.7 Forced termination by software

After DSTn is set to 0 by software, it takes up to 2 clocks until a DMA transfer is actually stopped and DSTn is set to 0. To forcibly terminate a DMA transfer by software without waiting for occurrence of the interrupt (INTDMAn) of DMAn, therefore, perform either of the following processes.

<When using one DMA channel>

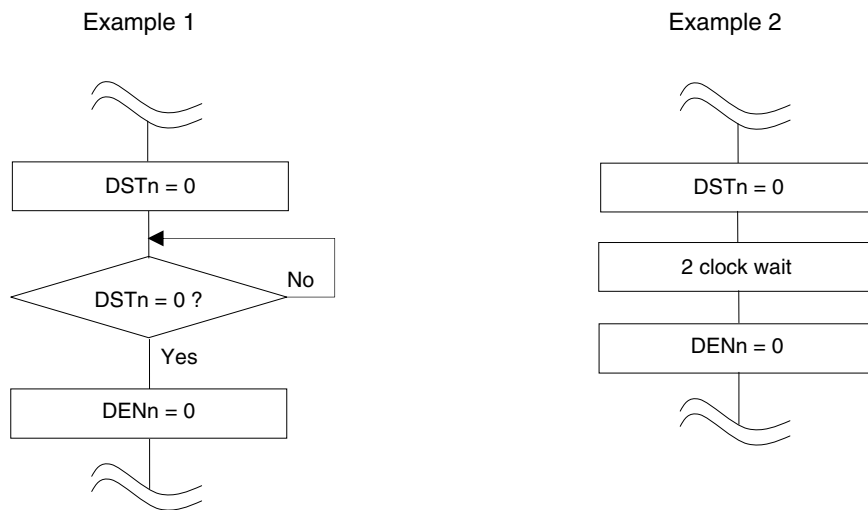
- Set DSTn to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software, confirm by polling that DSTn has actually been cleared to 0, and then set DENn to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction).
- Set DSTn to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software and then set DENn to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction) two or more clocks after.

<R>

<When using both DMA channels>

- To forcibly terminate DMA transfer by software when using both DMA channels (by setting DSTn to 0), clear the DSTn bit to 0 after the DMA transfer is held pending by setting the DWAIT0 and DWAIT1 bits of both channels to 1. Next, clear the DWAIT0 and DWAIT1 bits of both channels to 0 to cancel the pending status, and then clear the DENn bit to 0.

**Figure 16-13. Forced Termination of DMA Transfer (1/2)**



- Remarks**
1. n: DMA channel number (n = 0, 1)
  2. 1 clock:  $1/f_{CLK}$  ( $f_{CLK}$ : CPU clock)

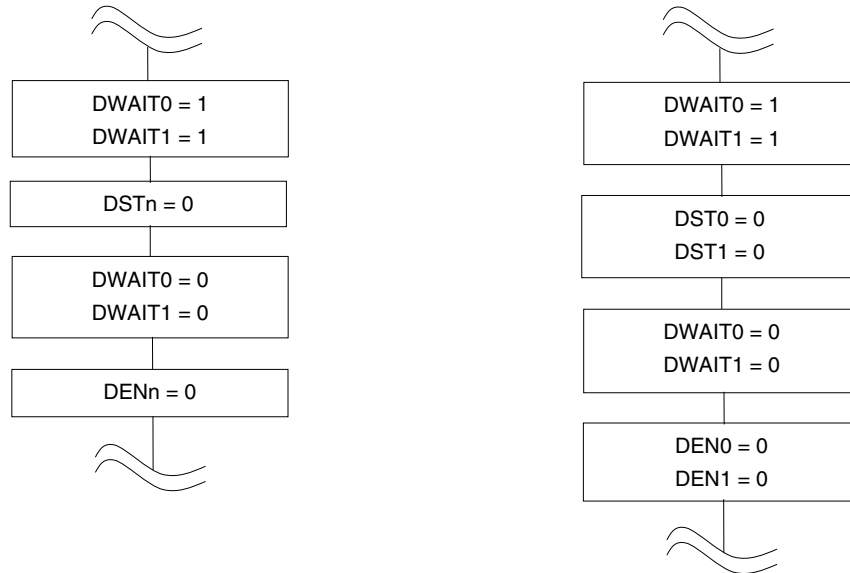
Figure 16-13. Forced Termination of DMA Transfer (2/2)

## Example 3

&lt;R&gt;

- Procedure for forcibly terminating the DMA transfer for one channel if both channels are used

- Procedure for forcibly terminating the DMA transfer for both channels if both channels are used



**Caution** In example 3, the system is not required to wait two clock cycles after  $DWAITn$  is set to 1. In addition, the system does not have to wait two clock cycles after clearing  $DSTn$  to 0, because more than two clock cycles elapse from when  $DSTn$  is cleared to 0 to when  $DENn$  is cleared to 0.

- Remarks**
1.  $n$ : DMA channel number ( $n = 0, 1$ )
  2. 1 clock:  $1/f_{CLK}$  ( $f_{CLK}$ : CPU clock)

## 16.6 Cautions on Using DMA Controller

- <R> **(1) Priority of DMA**
- During DMA transfer, a request from the other DMA channel is held pending even if generated. The pending DMA transfer is started after the ongoing DMA transfer is completed. If two DMA requests are generated at the same time, however, DMA channel 0 takes priority over DMA channel 1.
- If a DMA request and an interrupt request are generated at the same time, the DMA transfer takes precedence, and then interrupt servicing is executed.

&lt;R&gt;

**(2) DMA response time**

The response time of DMA transfer is as follows.

**Table 16-2. Response Time of DMA Transfer**

	Minimum Time	Maximum Time
Response time <sup>Note</sup>	3 clocks	10 clocks

**Note** This is the time required to execute an instruction from internal ROM (without accessing data in external memory). However, DMA transfers might be further delayed in the cases below. The number of clock cycles by which transfers are delayed varies depending on the condition.

Condition		Maximum Response Time
Executing an instruction from internal ROM	When external memory data is accessed	$8 + (3 \times (\text{external wait} + \text{internal wait}))$ clocks
Executing an instruction from internal RAM	When external memory data is not accessed	16 clocks
	When external memory data is accessed	$16 + (3 \times (\text{external wait} + \text{internal wait}))$ clocks
Executing an instruction from external memory		$16 + (12 \times (\text{external wait} + \text{internal wait}))$ clocks

- Cautions 1.** The above response time does not include the two clock cycles required for a DMA transfer.
- When executing a DMA pending instruction (see 16.6 (4)), the maximum response time is extended by the execution time of that instruction to be held pending.
  - Do not specify successive transfer triggers for a channel within a period equal to the maximum response time plus one clock cycle, because they might be ignored.

**Remarks 1.** Internal Wait : Number of clock cycles the system waits according to the clock selected for the CLKOUT pin of the external bus interface

CLKOUT Pin Selection Clock	Number of Wait States
f <sub>CLK</sub>	3 clocks
f <sub>CLK</sub> /2	5 to 6 clocks
f <sub>CLK</sub> /3	7 to 9 clocks
f <sub>CLK</sub> /4	9 to 12 clocks

- External wait: Low level period of the  $\overline{\text{WAIT}}$  pin of the external bus interface (in  $1/f_{\text{CLK}}$  units). For details, see 5.6 **Number of Instruction Wait Clocks for External Wait Pin**.
- 1 clock:  $1/f_{\text{CLK}}$  ( $f_{\text{CLK}}$ : CPU clock)

**(3) Operation in standby mode**

The DMA controller operates as follows in the standby mode.

**Table 16-3. DMA Operation in Standby Mode**

Status	DMA Operation
HALT mode	Normal operation
STOP mode	Stops operation. If DMA transfer and STOP instruction execution contend, DMA transfer may be damaged. Therefore, stop DMA before executing the STOP instruction.

**(4) DMA pending instruction**

Even if a DMA request is generated, DMA transfer is held pending immediately after the following instructions.

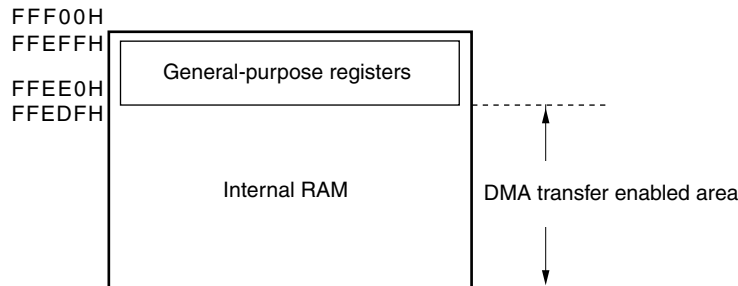
- CALL !addr16
- CALL \$!addr20
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- Bit manipulation instructions for registers IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H and PSW each.

**(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified**

The address indicated by DRA0n is incremented during DMA transfer. If the address is incremented to an address in the general-purpose register area or exceeds the area of the internal RAM, the following operation is performed.

- In mode of transfer from SFR to RAM  
The data of that address is lost.
- In mode of transfer from RAM to SFR  
Undefined data is transferred to SFR.

In either case, malfunctioning may occur or damage may be done to the system. Therefore, make sure that the address is within the internal RAM area other than the general-purpose register area.





## CHAPTER 17 INTERRUPT FUNCTIONS

### 17.1 Interrupt Function Types

The following two types of interrupt functions are used.

#### (1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the priority of vectored interrupt servicing. For the priority order, see **Table 17-1**.

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.

External: 13, internal: 32

#### (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.

### 17.2 Interrupt Sources and Configuration

The 78K0R/KF3 has a total of 46 interrupt sources including maskable interrupts and software interrupts. In addition, they also have up to five reset sources (see **Table 17-1**). 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 17-1. Interrupt Source List (1/3)

Interrupt Type	Default Priority <sup>Note 1</sup>	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type <sup>Note 2</sup>
		Name	Trigger			
Maskable	0	INTWDTI	Watchdog timer interval <sup>Note 3</sup> (75% of overflow time)	Internal	0004H	(A)
	1	INTLVI	Low-voltage detection <sup>Note 4</sup>		0006H	
	2	INTP0	Pin input edge detection	External	0008H	(B)
	3	INTP1			000AH	
	4	INTP2			000CH	
	5	INTP3			000EH	
	6	INTP4			0010H	
	7	INTP5			0012H	
	8	INTST3			UART3 transmission transfer end or buffer empty interrupt	
	9	INTSR3	UART3 reception transfer end	0016H		
	10	INTSRE3	UART3 reception communication error occurrence	0018H		
	11	INTDMA0	End of DMA0 transfer	001AH		
	12	INTDMA1	End of DMA1 transfer	001CH		
	13	INTST0 /INTCSI00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt	001EH		
	14	INTSR0 /INTCSI01	UART0 reception transfer end/ CSI01 transfer end or buffer empty interrupt	0020H		
	15	INTSRE0	UART0 reception communication error occurrence	0022H		
	16	INTST1 /INTCSI10 /INTIIC10	UART1 transmission transfer end or buffer empty interrupt/ CSI10 transfer end or buffer empty interrupt/ IIC10 transfer end	0024H		
	17	INTSR1 /INTCSI11 /INTIIC11	UART1 reception transfer end /CSI11 transfer end or buffer empty interrupt/IIC11 transfer end	0026H		
	18	INTSRE1	UART1 reception communication error occurrence	0028H		
	19	INTIIC0	End of IIC0 communication	002AH		
	20	INTTM00	End of timer array unit 0 channel 0 count or capture	002CH		
21	INTTM01	End of timer array unit 0 channel 1 count or capture	002EH			

- Notes**
1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 44 indicates the lowest priority.
  2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 17-1.
  3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.
  4. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is cleared to 0.

Table 17-1. Interrupt Source List (2/3)

Interrupt Type	Default Priority <sup>Note 1</sup>	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type <sup>Note 2</sup>
		Name	Trigger			
Maskable	22	INTTM02	End of timer array unit 0 channel 2 count or capture	Internal	0030H	(A)
	23	INTTM03	End of timer array unit 0 channel 3 count or capture		0032H	
	24	INTAD	End of A/D conversion		0034H	
	25	INTRTC	Fixed-cycle signal of real-time counter/alarm match detection		0036H	
	26	INTRTCI	Interval signal detection of real-time counter		0038H	
	27	INTKR	Key return signal detection		External	
	28	INTST2 /INTCSI20 /INTIIC20	UART2 transmission transfer end or buffer empty interrupt/CSI20 transfer end or buffer empty interrupt/IIC20 transfer end	Internal	003CH	(A)
	29	INTSR2 /INTCSI21 /INTIIC21	UART2 reception transfer end/CSI21 transfer end or buffer empty interrupt/IIC21 transfer end		003EH	
	30	INTSRE2	UART2 reception communication error occurrence		0040H	
	31	INTTM04	End of timer array unit 0 channel 4 count or capture		0042H	
	32	INTTM05	End of timer array unit 0 channel 5 count or capture		0044H	
	33	INTTM06	End of timer array unit 0 channel 6 count or capture		0046H	
	34	INTTM07	End of timer array unit 0 channel 7 count or capture		0048H	
	35	INTP6	Pin input edge detection	External	004AH	(D)
	36	INTP7			004CH	
	37	INTP8			004EH	
	38	INTP9			0050H	
	39	INTP10			0052H	
	40	INTP11			0054H	
	41	INTTM10	End of timer array unit 1 channel 0 count or capture	Internal	0056H	(A)
	42	INTTM11	End of timer array unit 1 channel 1 count or capture		0058H	
	43	INTTM12	End of timer array unit 1 channel 2 count or capture		005AH	
44	INTTM13	End of timer array unit 1 channel 3 count or capture	005CH			

- Notes**
1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 44 indicates the lowest priority.
  2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 17-1.

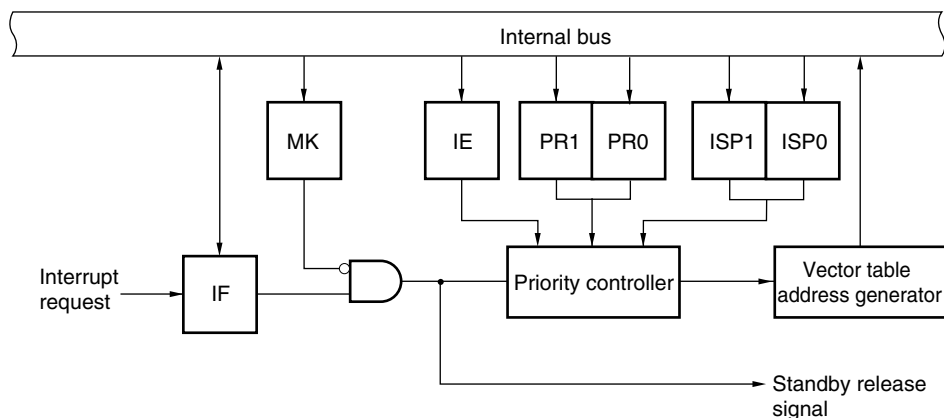
Table 17-1. Interrupt Source List (3/3)

Interrupt Type	Default Priority <sup>Note 1</sup>	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type <sup>Note 2</sup>
		Name	Trigger			
Software	–	BRK	Execution of BRK instruction	–	007EH	(D)
Reset	–	RESET	$\overline{\text{RESET}}$ pin input	–	0000H	–
		POC	Power-on-clear			
		LVI	Low-voltage detection <sup>Note 3</sup>			
		WDT	Overflow of watchdog timer			
		TRAP	Execution of illegal instruction <sup>Note 4</sup>			

- Notes**
1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 44 indicates the lowest priority.
  2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 17-1.
  3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.
  4. When the instruction code in FFH is executed.  
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

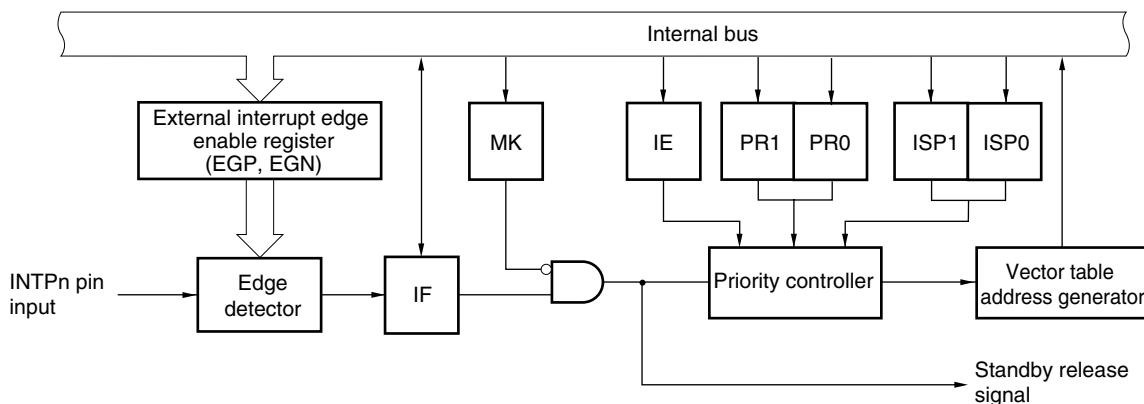
Figure 17-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



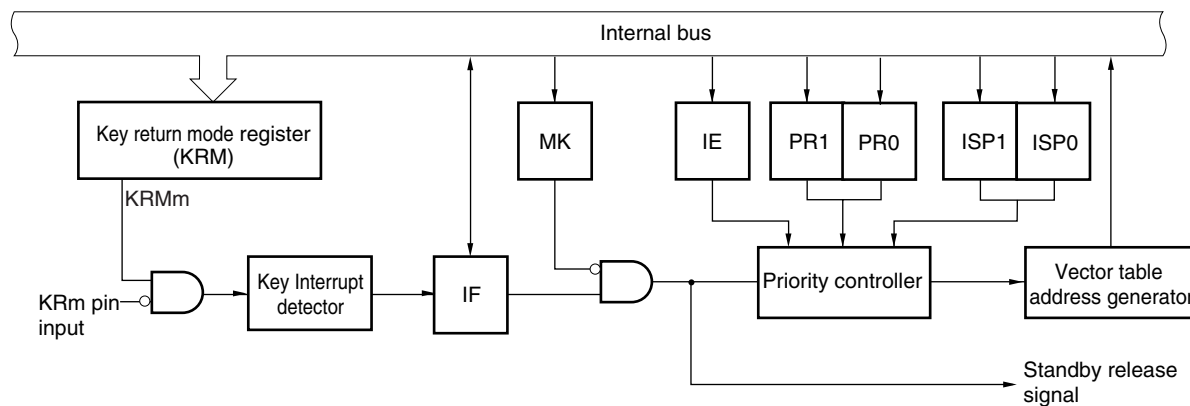
<R>

(B) External maskable interrupt (INTPn)

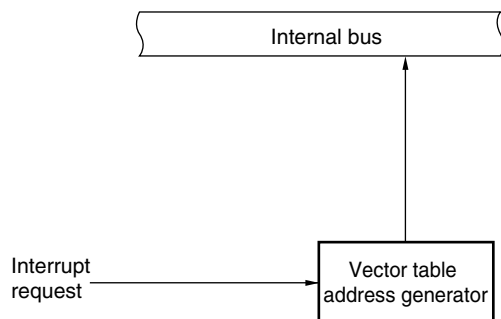


- Remarks 1.** IF: Interrupt request flag  
 IE: Interrupt enable flag  
 ISP0: In-service priority flag 0  
 ISP1: In-service priority flag 1  
 MK: Interrupt mask flag  
 PR0: Priority specification flag 0  
 PR1: Priority specification flag 1
- 2.** n = 0 to 11

<R> (C) External maskable interrupt (INTKR)



(D) Software interrupt



- Remarks 1.** IF: Interrupt request flag  
 IE: Interrupt enable flag  
 ISP0: In-service priority flag 0  
 ISP1: In-service priority flag 1  
 MK: Interrupt mask flag  
 PR0: Priority specification flag 0  
 PR1: Priority specification flag 1  
**2.** m = 0 to 7

### 17.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)
- External interrupt rising edge enable registers (EGP0, EGP1)
- External interrupt falling edge enable registers (EGN0, EGN1)
- Program status word (PSW)

Table 17-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

**Table 17-2. Flags Corresponding to Interrupt Request Sources (1/2)**

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
	Register	Register	Register	Register	Register	Register
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L
INTLVI	LVIIIF		LVIMK		LVIPR0, LVIPR1	
INTP0	PIF0		PMK0		PPR00, PPR10	
INTP1	PIF1		PMK1		PPR01, PPR11	
INTP2	PIF2		PMK2		PPR02, PPR12	
INTP3	PIF3		PMK3		PPR03, PPR13	
INTP4	PIF4		PMK4		PPR04, PPR14	
INTP5	PIF5		PMK5		PPR05, PPR15	
INTST3	STIF3	IF0H	STMK3	MK0H	STPR03, STPR13	PR00H, PR10H
INTSR3	SRIF3		SRMK3		SRPR03, SRPR13	
INTSRE3	SREIF3		SREMK3		SREPR03, SREPR13	
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10	
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11	
INTST0 <small>Note 1</small>	STIF0 <small>Note 1</small>		STMK0 <small>Note 1</small>		STPR00, STPR10 <small>Note 1</small>	
INTCSI00 <small>Note 1</small>	CSIIIF00 <small>Note 1</small>		CSIMK00 <small>Note 1</small>		CSIPR000, CSIPR100 <small>Note 1</small>	
INTSR0 <small>Note 2</small>	SRIF0 <small>Note 2</small>		SRMK0 <small>Note 2</small>		SRPR00, SRPR10 <small>Note 2</small>	
INTCSI01 <small>Note 2</small>	CSIIIF01 <small>Note 2</small>		CSIMK01 <small>Note 2</small>		CSIPR001, CSIPR101 <small>Note 2</small>	
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10	

- Notes**
1. Do not use UART0 and CSI00 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST0 and INTCSI00 is generated, bit 5 of IF1H is set to 1. Bit 5 of MK0H, PR00H, and PR10H supports these two interrupt sources.
  2. Do not use UART0 and CSI01 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTSR0 and INTCSI01 is generated, bit 6 of IF0H is set to 1. Bit 6 of MK0H, PR00H, and PR10H supports these two interrupt sources.

Table 17-2. Flags Corresponding to Interrupt Request Sources (2/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
	Register	Register	Register	Register	Register	Register
INTST1 <sup>Note 1</sup>	STIF1 <sup>Note 1</sup>	IF1L	STMK1 <sup>Note 1</sup>	MK1L	STPR01, STPR11 <sup>Note 1</sup>	PR01L, PR11L
INTCSI10 <sup>Note 1</sup>	CSIF10 <sup>Note 1</sup>		CSIMK10 <sup>Note 1</sup>		CSIPR010, CSIPR110 <sup>Note 1</sup>	
INTIIC10 <sup>Note 1</sup>	IICIF10 <sup>Note 1</sup>		IICMK10 <sup>Note 1</sup>		IICPR010, IICPR110 <sup>Note 1</sup>	
INTSR1 <sup>Note 2</sup>	SRIF1 <sup>Note 2</sup>		SRMK1 <sup>Note 2</sup>		SRPR01, SRPR11 <sup>Note 2</sup>	
INTCSI11 <sup>Note 2</sup>	CSIF11 <sup>Note 2</sup>		CSIMK11 <sup>Note 2</sup>		CSIPR011, CSIPR111 <sup>Note 2</sup>	
INTIIC11 <sup>Note 2</sup>	IICIF11 <sup>Note 2</sup>		IICMK11 <sup>Note 2</sup>		IICPR011, IICPR111 <sup>Note 2</sup>	
INTSRE1	SREIF1		SREMK1		SREPR01, SREPR11	
INTIIC0	IICIF0		IICMK0		IICPR00, IICPR10	
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102	
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103	
INTAD	ADIF		IF1H		ADMK	
INTRTC	RTCIF	RTCMK		RT CPR0, RT CPR1		
INTRTCI	RTCIF	RTCIMK		RT CIPR0, RT CIPR1		
INTKR	KRIF	KRMK		KRPR0, KRPR1		
INTST2 <sup>Note 3</sup>	STIF2 <sup>Note 3</sup>	STMK2 <sup>Note 3</sup>		STPR02, STPR12 <sup>Note 3</sup>		
INTCSI20 <sup>Note 3</sup>	CSIF20 <sup>Note 3</sup>	CSIMK20 <sup>Note 3</sup>		CSIPR020, CSIPR120 <sup>Note 3</sup>		
INTIIC20 <sup>Note 3</sup>	IICIF20 <sup>Note 3</sup>	IICMK20 <sup>Note 3</sup>		IICPR020, IICPR120 <sup>Note 3</sup>		
INTSR2 <sup>Note 4</sup>	SRIF2 <sup>Note 4</sup>	SRMK2 <sup>Note 4</sup>		SRPR02, SRPR12 <sup>Note 4</sup>		
INTCSI21 <sup>Note 4</sup>	CSIF21 <sup>Note 4</sup>	CSIMK21 <sup>Note 4</sup>		CSIPR021, CSIPR121 <sup>Note 4</sup>		
INTIIC21 <sup>Note 4</sup>	IICIF21 <sup>Note 4</sup>	IICMK21 <sup>Note 4</sup>		IICPR021, IICPR121 <sup>Note 4</sup>		
INTSRE2	SREIF2	SREMK2		SREPR02, SREPR12		
INTTM04	TMIF04	TMMK04		TMPR004, TMPR104		

- Notes**
1. Do not use UART1, CSI10, and IIC10 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST1, INTCSI10, and INTIIC10 is generated, bit 0 of IF1L is set to 1. Bit 0 of MK1L, PR01L, and PR11L supports these three interrupt sources.
  2. Do not use UART1, CSI11, and IIC11 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST1, INTCSI11, and INTIIC11 is generated, bit 1 of IF1L is set to 1. Bit 1 of MK1L, PR01L, and PR11L supports these three interrupt sources.
  3. Do not use UART2, CSI20, and IIC20 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST2, INTCSI20, and INTIIC20 is generated, bit 4 of IF1H is set to 1. Bit 4 of MK1H, PR01H, and PR11H supports these three interrupt sources.
  4. Do not use UART2, CSI21, and IIC21 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST2, INTCSI21, and INTIIC21 is generated, bit 5 of IF1H is set to 1. Bit 5 of MK1H, PR01H, and PR11H supports these three interrupt sources.



**Table 17-2. Flags Corresponding to Interrupt Request Sources (3/3)**

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag				
		Register		Register		Register			
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L			
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106				
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107				
INTP6	PIF6		PMK6		PPR06, PPR16				
INTP7	PIF7		PMK7		PPR07, PPR17				
INTP8	PIF8		PMK8		PPR08, PPR18				
INTP9	PIF9		PMK9		PPR09, PPR19				
INTP10	PIF10		PMK10		PPR010, PPR110				
INTP11	PIF11		IF2H		PMK11		MK2H	PPR011, PPR111	PR02H, PR12H
INTTM10	TMIF10				TMMK10			TMPR010, TMPR110	
INTTM11	TMIF11	TMMK11		TMPR011, TMPR111					
INTTM12	TMIF12	TMMK12		TMPR012, TMPR112					
INTTM13	TMIF13	TMMK13		TMPR013, TMPR113					

**(1) Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)**

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

IF0L, IF0H, IF1L, IF1H, IF2L, and IF2H can be set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H, IF1L and IF1H, and IF2L and IF2H are combined to form 16-bit registers IF0, IF1, and IF2, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Remark** If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

**Figure 17-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (1/2)**

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0	SRIF0 CSIIF01	STIF0 CSIIF00	DMAIF1	DMAIF0	SREIF3	SRIF3	STIF3

Figure 17-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (2/2)

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	IICIF0	SREIF1	SRIF1 CSIF11 IICIF11	STIF1 CSIF10 IICIF10

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	TMIF04	SREIF2	SRIF2 CSIF21 IICIF21	STIF2 CSIF20 IICIF20	KRIF	RTCIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	PIF10	PIF9	PIF8	PIF7	PIF6	TMIF07	TMIF06	TMIF05

Address: FFFD1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
IF2H	0	0	0	TMIF13	TMIF12	TMIF11	TMIF10	PIF11

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

- Cautions**
1. Be sure to clear bits 5 to 7 of IF2H to 0.
  2. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.
  3. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "\_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

**(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)**

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H can be set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, MK1L and MK1H, and MK2L and MK2H are combined to form 16-bit registers MK0, MK1, and MK2, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

**Remark** If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

**Figure 17-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)**

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0	SRMK0 CSIMK01	STMK0 CSIMK00	DMAMK1	DMAMK0	SREMK3	SRMK3	STMK3

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	IICMK0	SREMK1	SRMK1 CSIMK11 IICMK11	STMK1 CSIMK10 IICMK10

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	TMMK04	SREMK2	SRMK2 CSIMK21 IICMK21	STMK2 CSIMK20 IICMK20	KRMK	RTCIMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	PMK10	PMK9	PMK8	PMK7	PMK6	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
MK2H	1	1	1	TMMK13	TMMK12	TMMK11	TMMK10	PMK11

XXMKX	Interrupt servicing control						
0	Interrupt servicing enabled						
1	Interrupt servicing disabled						

**Caution** Be sure to set bits 5 to 7 of MK2H to 1.

**(3) Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)**

The priority specification flag registers are used to set the corresponding maskable interrupt priority level. A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L, or 2H). PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H can be set by a 1-bit or 8-bit memory manipulation instruction. If PR00L and PR00H, PR01L and PR01H, PR02L and PR02H, PR10L and PR10H, PR11L and PR11H, and PR12L and PR12H are combined to form 16-bit registers PR00, PR01, PR02, PR10, PR11, and PR12, they can be set by a 16-bit memory manipulation instruction. Reset signal generation sets these registers to FFH.

**Remark** If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

**Figure 17-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (1/2)**

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00	SRPR00	STPR00	DMAPR01	DMAPR00	SREPR03	SRPR03	STPR03
		CSIPR001	CSIPR000					

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10	SRPR10	STPR10	DMAPR11	DMAPR10	SREPR13	SRPR13	STPR13
		CSIPR101	CSIPR100					

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICPR00	SREPR01	SRPR01	STPR01
							CSIPR011	CSIPR010
							IICPR011	IICPR010

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICPR10	SREPR11	SRPR11	STPR11
							CSIPR111	CSIPR110
							IICPR111	IICPR110

**Figure 17-4. Format of Priority Specification Flag Registers  
(PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (2/2)**

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	TMPR004	SREPR02	SRPR02 CSIPR021 IICPR021	STPR02 CSIPR020 IICPR020	KRPR0	RTCIPR0	RTCPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	TMPR104	SREPR12	SRPR12 CSIPR121 IICPR121	STPR12 CSIPR120 IICPR120	KRPR1	RTCIPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	PPR010	PPR09	PPR08	PPR07	PPR06	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	PPR110	PPR19	PPR18	PPR17	PPR16	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
PR02H	1	1	1	TMPR013	TMPR012	TMPR011	TMPR010	PPR011

Address: FFFDDH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
PR12H	1	1	1	TMPR113	TMPR112	TMPR111	TMPR110	PPR111

XXPR1X	XXPR0X	Priority level selection
0	0	Specify level 0 (high priority level)
0	1	Specify level 1
1	0	Specify level 2
1	1	Specify level 3 (low priority level)

**Caution** Be sure to set bits 5 to 7 of PR02H and PR12H to 1.

**(4) External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)**

These registers specify the valid edge for INTPO to INTP11.

EGP0, EGP1, EGN0, and EGN1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

**Figure 17-5. Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1)**

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP11	EGP10	EGP9	EGP8

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN11	EGN10	EGN9	EGN8

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 11)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 17-3 shows the ports corresponding to EGPn and EGNn.

**Table 17-3. Ports Corresponding to EGPn and EGNn**

Detection Enable Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P46	INTP1
EGP2	EGN2	P47	INTP2
EGP3	EGN3	P30	INTP3
EGP4	EGN4	P31	INTP4
EGP5	EGN5	P16	INTP5
EGP6	EGN6	P140	INTP6
EGP7	EGN7	P141	INTP7
EGP8	EGN8	P74	INTP8
EGP9	EGN9	P75	INTP9
EGP10	EGN10	P76	INTP10
EGP11	EGN11	P77	INTP11

**Caution** Select the port mode by clearing EGPn and EGNn to 0 because an edge may be detected when the external interrupt function is switched to the port function.

**Remark** n = 0 to 11

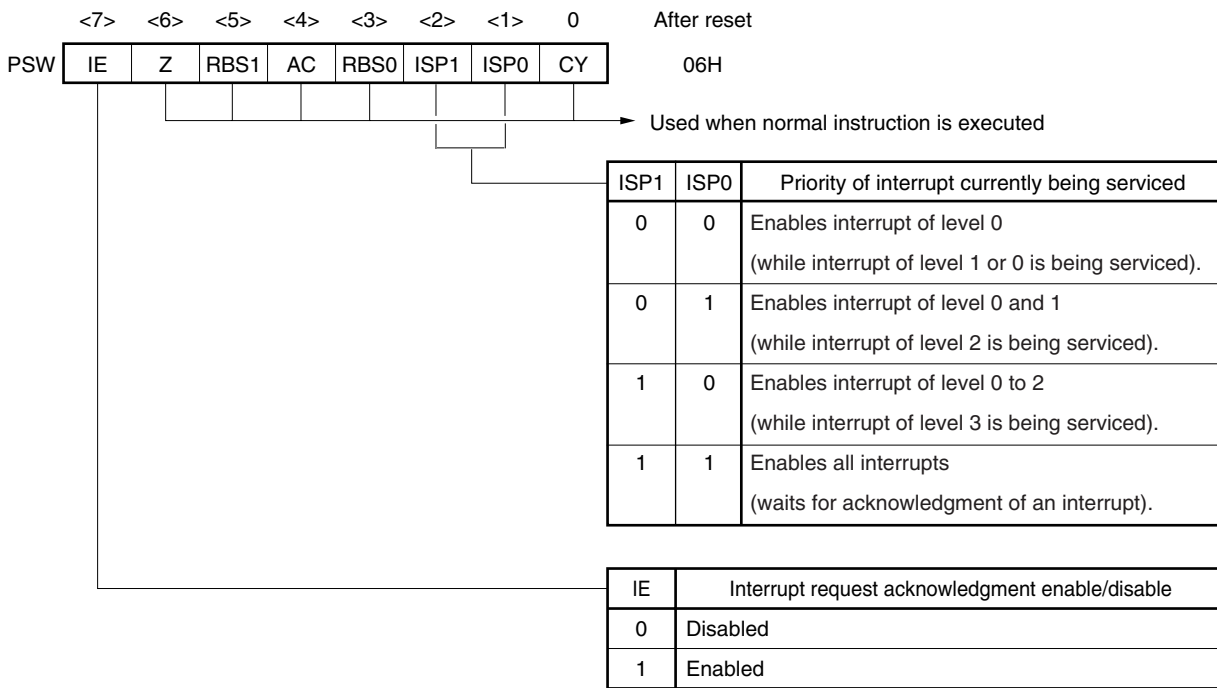
**(5) Program status word (PSW)**

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 17-6. Configuration of Program Status Word**





## 17.4 Interrupt Servicing Operations

### 17.4.1 Maskable interrupt acknowledgment

A maskable interrupt becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request.

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 17-4 below.

For the interrupt request acknowledgment timing, see **Figures 17-8** and **17-9**.

**Table 17-4. Time from Generation of Maskable Interrupt Until Servicing**

	Minimum Time	Maximum Time <sup>Note</sup>
Servicing time	9 clocks	14 clocks

**Note** If an interrupt request is generated just before the RET instruction, the wait time becomes longer.

**Remark** 1 clock:  $1/f_{\text{CLK}}$  ( $f_{\text{CLK}}$ : 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.

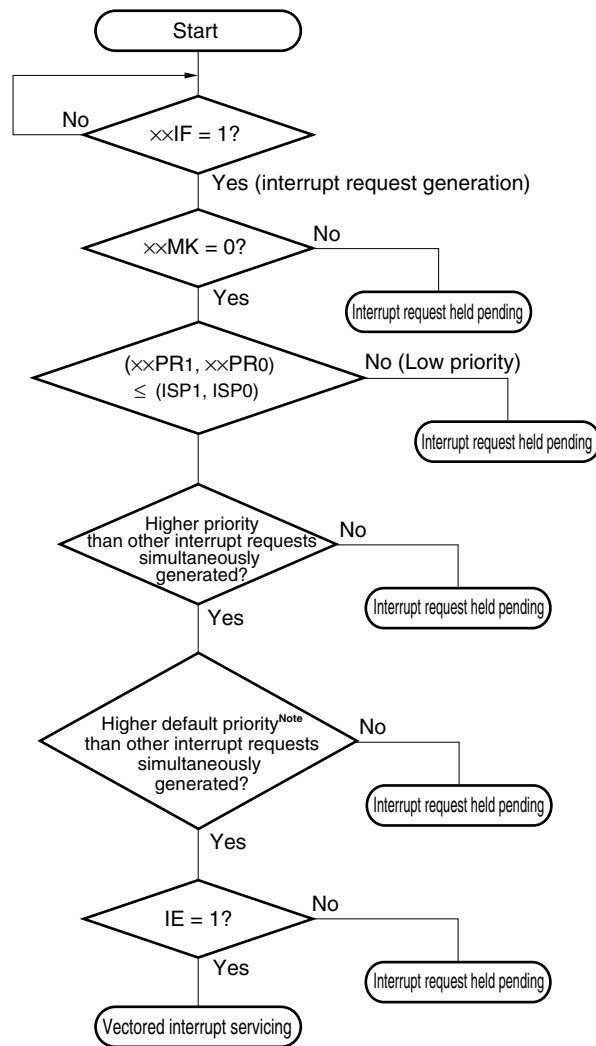
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 17-7 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 ISPO 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 17-7. Interrupt Request Acknowledgment Processing Algorithm



xxIF: Interrupt request flag

xxMK: Interrupt mask flag

xxPR0: Priority specification flag 0

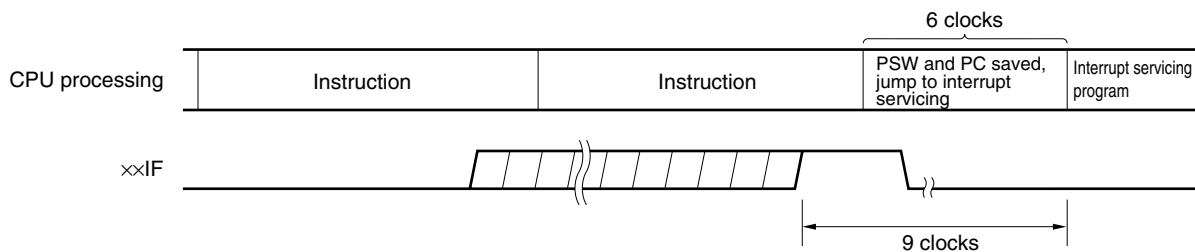
xxPR1: Priority specification flag 1

IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)

ISP0, ISP1: Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 17-6**)

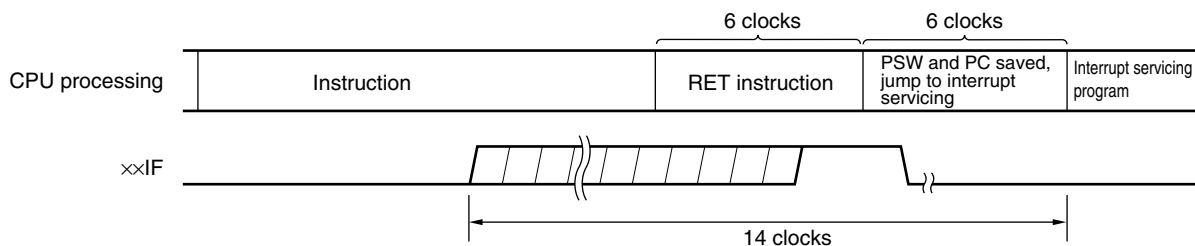
**Note** For the default priority, refer to **Table 17-1 Interrupt Source List**.

**Figure 17-8. Interrupt Request Acknowledgment Timing (Minimum Time)**



**Remark** 1 clock:  $1/f_{CLK}$  ( $f_{CLK}$ : CPU clock)

**Figure 17-9. Interrupt Request Acknowledgment Timing (Maximum Time)**



**Remark** 1 clock:  $1/f_{CLK}$  ( $f_{CLK}$ : CPU clock)

#### 17.4.2 Software interrupt request acknowledgment

A software interrupt acknowledge is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (0007EH, 0007FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

**Caution** Do not use the RETI instruction for restoring from the software interrupt.

**17.4.3 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.

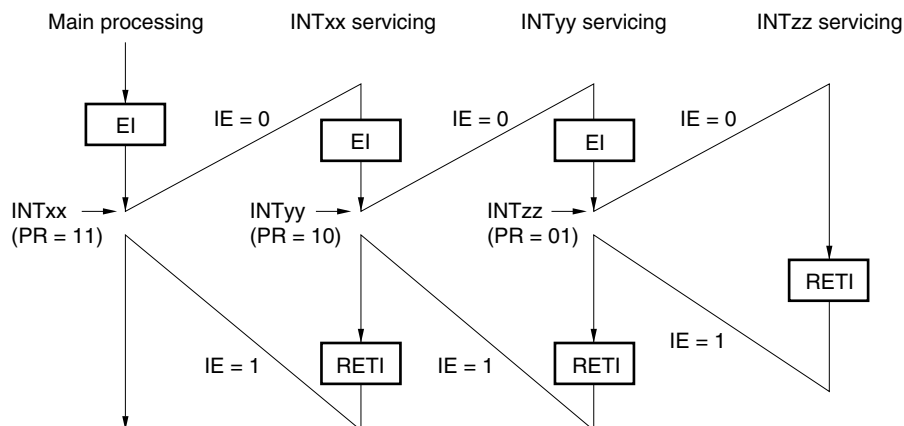
Table 17-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 17-10 shows multiple interrupt servicing examples.

**Table 17-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing**

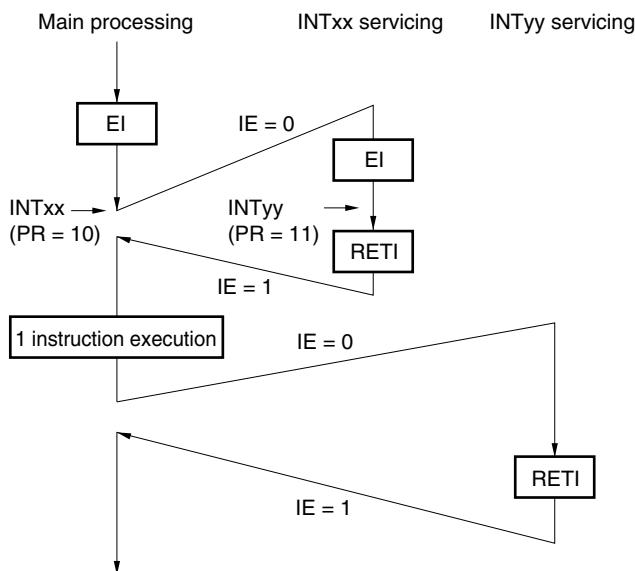
Multiple Interrupt Request Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0 ISP0 = 0	○	×	×	×	×	×	×	×	○
	ISP1 = 0 ISP0 = 1	○	×	○	×	×	×	×	×	○
	ISP1 = 1 ISP0 = 0	○	×	○	×	○	×	×	×	○
	ISP1 = 1 ISP0 = 1	○	×	○	×	○	×	○	×	○
Software interrupt		○	×	○	×	○	×	○	×	○

- Remarks**
- : Multiple interrupt servicing enabled
  - ×: Multiple interrupt servicing disabled
  - 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.
  - PR is a flag contained in PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H.
    - PR = 00: Specify level 0 with  $\text{xxPR1x} = 0, \text{xxPR0x} = 0$  (higher priority level)
    - PR = 01: Specify level 1 with  $\text{xxPR1x} = 0, \text{xxPR0x} = 1$
    - PR = 10: Specify level 2 with  $\text{xxPR1x} = 1, \text{xxPR0x} = 0$
    - PR = 11: Specify level 3 with  $\text{xxPR1x} = 1, \text{xxPR0x} = 1$  (lower priority level)

Figure 17-10. 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  $\times\times PR1\times = 0$ ,  $\times\times PR0\times = 0$  (higher priority level)

PR = 01: Specify level 1 with  $\times\times PR1\times = 0$ ,  $\times\times PR0\times = 1$

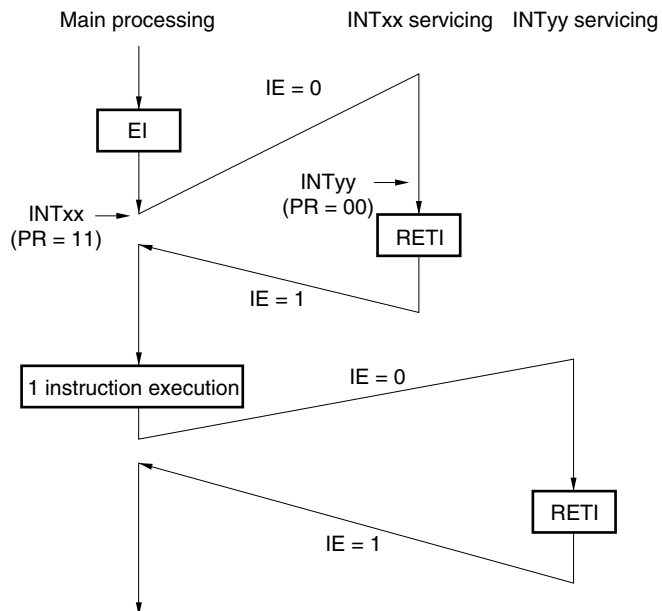
PR = 10: Specify level 2 with  $\times\times PR1\times = 1$ ,  $\times\times PR0\times = 0$

PR = 11: Specify level 3 with  $\times\times PR1\times = 1$ ,  $\times\times PR0\times = 1$  (lower priority level)

IE = 0: Interrupt request acknowledgment is disabled

IE = 1: Interrupt request acknowledgment is enabled.

Figure 17-10. 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  $\times\times PR1\times = 0$ ,  $\times\times PR0\times = 0$  (higher priority level)

PR = 01: Specify level 1 with  $\times\times PR1\times = 0$ ,  $\times\times PR0\times = 1$

PR = 10: Specify level 2 with  $\times\times PR1\times = 1$ ,  $\times\times PR0\times = 0$

PR = 11: Specify level 3 with  $\times\times PR1\times = 1$ ,  $\times\times PR0\times = 1$  (lower priority level)

IE = 0: Interrupt request acknowledgment is disabled

IE = 1: Interrupt request acknowledgment is enabled.

**17.4.4 Interrupt request hold**

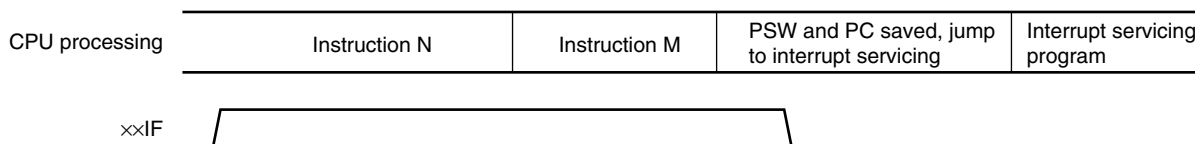
There are instructions where, even if an interrupt request is issued while the instruction are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- <R> • BTCLR PSW. bit, \$addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- <R> • SKH
- <R> • SKNH
- Manipulation instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers.

**Caution** The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged.

Figure 17-11 shows the timing at which interrupt requests are held pending.

**Figure 17-11. Interrupt Request Hold**



- Remarks**
1. Instruction N: Interrupt request hold instruction
  2. Instruction M: Instruction other than interrupt request hold instruction
  3. The xxPR (priority level) values do not affect the operation of xxIF (interrupt request).

## CHAPTER 18 KEY INTERRUPT FUNCTION

### 18.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by setting the key return mode register (KRM) and inputting a falling edge to the key interrupt input pins (KR0 to KR7).

**Table 18-1. Assignment of Key Interrupt Detection Pins**

Flag	Description
KRM0	Controls KR0 signal in 1-bit units.
KRM1	Controls KR1 signal in 1-bit units.
KRM2	Controls KR2 signal in 1-bit units.
KRM3	Controls KR3 signal in 1-bit units.
KRM4	Controls KR4 signal in 1-bit units.
KRM5	Controls KR5 signal in 1-bit units.
KRM6	Controls KR6 signal in 1-bit units.
KRM7	Controls KR7 signal in 1-bit units.

### 18.2 Configuration of Key Interrupt

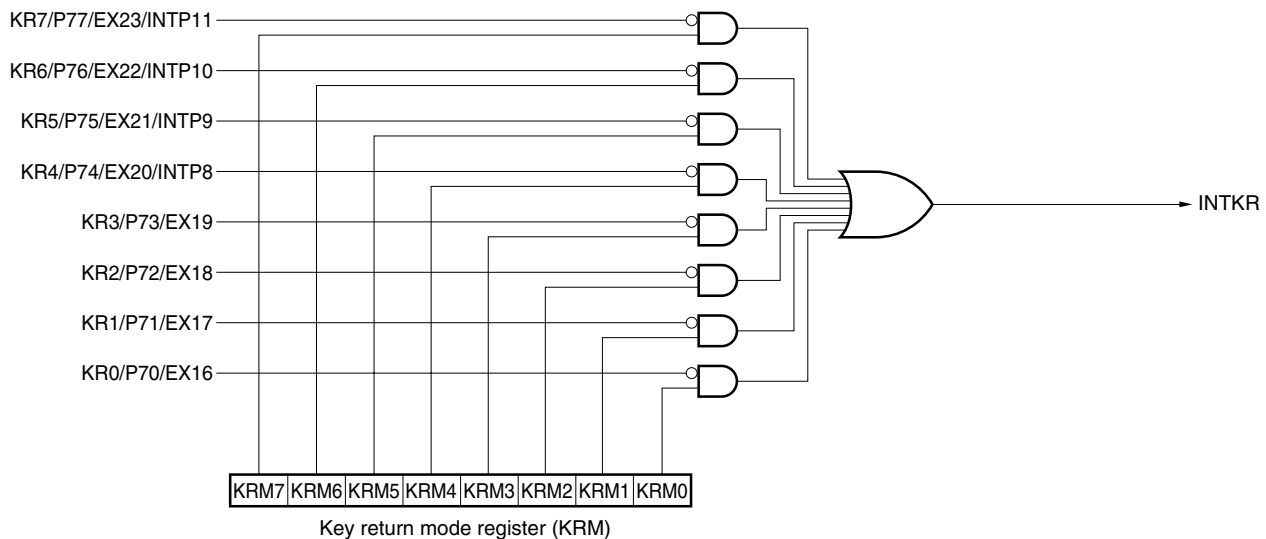
The key interrupt includes the following hardware.

**Table 18-2. Configuration of Key Interrupt**

Item	Configuration
Control register	Key return mode register (KRM) Port mode register 7 (PM7)

<R>

**Figure 18-1. Block Diagram of Key Interrupt**





### 18.3 Register Controlling Key Interrupt

#### (1) Key return mode register (KRM)

This register controls the KRM0 to KRM7 bits using the KR0 to KR7 signals, respectively.

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

Reset signal generation clears this register to 00H.

**Figure 18-2. Format of Key Return Mode Register (KRM)**

Address: FFF37H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

- Cautions**
1. If any of the KRM0 to KRM7 bits used is set to 1, set bits 0 to 7 (PU70 to PU77) of the corresponding pull-up resistor register 7 (PU7) to 1.
  2. An interrupt will be generated if the target bit of the KRM register is set while a low level is being input to the key interrupt input pin. To ignore this interrupt, set the KRM register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input low-level width (250 ns or more).
  3. The bits not used in the key interrupt mode can be used as normal ports.

**Remark**    n = 0 to 7

<R>

#### (2) Port mode register 7 (PM7)

This register sets the input or output of port 7 in 1-bit units.

When using the P70/KR0/EX16, P71/KR1/EX17, P72/KR2/EX18, P73/KR3/EX19, P74/KR4/EX20/INTP8, P75/KR5/EX21/INTP9, P76/KR6/EX22/INTP10, P77/KR7/EX23/INTP11 pins as the key interrupt function, set both PM70 to PM77 to 1. The output latches of P70 to P77 at this time may be 0 or 1.

PM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

**Figure 18-3. Format of Port Mode Register 7 (PM7)**

Address: FFF27H    After reset: FFH    R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

**Remark**    n = 0 to 7

## CHAPTER 19 STANDBY FUNCTION

### 19.1 Standby Function and Configuration

#### 19.1.1 Standby function

The standby function reduces the operating current of the system, and the following two modes are available.

##### (1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator, internal high-speed oscillator, or subsystem clock oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

##### (2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and internal high-speed oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

- Cautions**
1. The STOP mode can be used only when the CPU is operating on the main system clock. The STOP mode cannot be set while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.
  2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.
  3. 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 the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.
  4. It can be selected by the option byte whether the internal low-speed oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 24 OPTION BYTE.

#### 19.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

**Remark** For the registers that start, stop, or select the clock, see CHAPTER 6 CLOCK GENERATOR.

**(1) Oscillation stabilization time counter status register (OSTC)**

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

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

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

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

When reset is released (reset by  $\overline{\text{RESET}}$  input, POC, LVI, WDT, and executing an illegal instruction), the STOP instruction and MSTOP (bit 7 of CSC register) = 1 clear this register to 00H.

**Figure 19-1. Format of Oscillation Stabilization Time Counter Status Register (OSTC)**

Address: FFFA2H After reset: 00H R

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

MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18	Oscillation stabilization time status		
								$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$	
0	0	0	0	0	0	0	0	$2^9/f_x \text{ max.}$	25.6 $\mu\text{s}$ max.	12.8 $\mu\text{s}$ max.
1	0	0	0	0	0	0	0	$2^9/f_x \text{ min.}$	25.6 $\mu\text{s}$ min.	12.8 $\mu\text{s}$ min.
1	1	0	0	0	0	0	0	$2^9/f_x \text{ min.}$	51.2 $\mu\text{s}$ min.	25.6 $\mu\text{s}$ min.
1	1	1	0	0	0	0	0	$2^{10}/f_x \text{ min.}$	102.4 $\mu\text{s}$ min.	51.2 $\mu\text{s}$ min.
1	1	1	1	0	0	0	0	$2^{11}/f_x \text{ min.}$	204.8 $\mu\text{s}$ min.	102.4 $\mu\text{s}$ min.
1	1	1	1	1	0	0	0	$2^{13}/f_x \text{ min.}$	819.2 $\mu\text{s}$ min.	409.6 $\mu\text{s}$ min.
1	1	1	1	1	1	0	0	$2^{15}/f_x \text{ min.}$	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	$2^{17}/f_x \text{ min.}$	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	$2^{18}/f_x \text{ min.}$	26.21 ms min.	13.11 ms min.

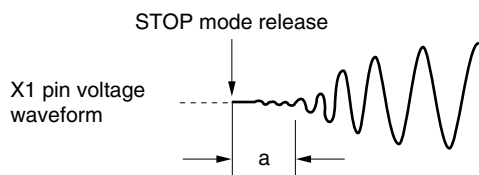
**Cautions 1.** After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.

**2.** The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTC. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- Desired OSTC oscillation stabilization time  $\leq$  Oscillation stabilization time set by OSTC

**Note,** therefore, that only the status up to the oscillation stabilization time set by OSTC is set to OSTC after STOP mode is released.

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



**Remark**  $f_x$ : X1 clock oscillation frequency

**(2) Oscillation stabilization time select register (OSTS)**

This register is used to select the X1 clock oscillation stabilization wait time when the STOP mode is released. When the X1 clock is selected as the CPU clock, the operation waits for the time set using OSTS after the STOP mode is released.

When the internal high-speed oscillation clock is selected as the CPU clock, confirm with OSTC that the desired oscillation stabilization time has elapsed after the STOP mode is released. The oscillation stabilization time can be checked up to the time set using OSTC.

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

Reset signal generation sets this register to 07H.

**Figure 19-2. Format of Oscillation Stabilization Time Select Register (OSTS)**

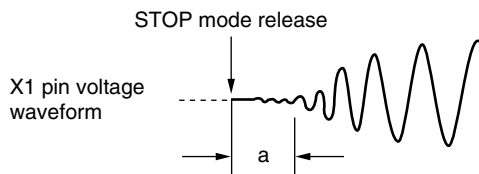
Address: FFFA3H After reset: 07H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				$f_x = 10 \text{ MHz}$	$f_x = 20 \text{ MHz}$
0	0	0	$2^8/f_x$	25.6 $\mu\text{s}$	Setting prohibited
0	0	1	$2^9/f_x$	51.2 $\mu\text{s}$	25.6 $\mu\text{s}$
0	1	0	$2^{10}/f_x$	102.4 $\mu\text{s}$	51.2 $\mu\text{s}$
0	1	1	$2^{11}/f_x$	204.8 $\mu\text{s}$	102.4 $\mu\text{s}$
1	0	0	$2^{13}/f_x$	819.2 $\mu\text{s}$	409.6 $\mu\text{s}$
1	0	1	$2^{15}/f_x$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_x$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_x$	26.21 ms	13.11 ms

- Cautions**
1. To set the STOP mode when the X1 clock is used as the CPU clock, set OSTS before executing the STOP instruction.
  2. Setting the oscillation stabilization time to 20  $\mu\text{s}$  or less is prohibited.
  3. Before changing the setting of the OSTS register, confirm that the count operation of the OSTC register is completed.
  4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.
  5. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.
    - Desired OSTC oscillation stabilization time  $\leq$  Oscillation stabilization time set by OSTS

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.
  6. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).



**Remark**  $f_x$ : X1 clock oscillation frequency

## 19.2 Standby Function Operation

### 19.2.1 HALT mode

#### (1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock, internal high-speed oscillation clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

**Table 19-1. Operating Statuses in HALT Mode (1/2)**

HALT Mode Setting		When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
		When CPU Is Operating on Internal High-Speed Oscillation Clock ( $f_{IH}$ )	When CPU Is Operating on X1 Clock ( $f_x$ )	When CPU Is Operating on External Main System Clock ( $f_{EX}$ )
Item				
System clock		Clock supply to the CPU is stopped		
Main system clock	$f_{IH}$	Operation continues (cannot be stopped)	Status before HALT mode was set is retained	
	$f_x$	Status before HALT mode was set is retained	Operation continues (cannot be stopped)	Cannot operate
	$f_{EX}$		Cannot operate	Operation continues (cannot be stopped)
Subsystem clock	$f_{XT}$	Status before HALT mode was set is retained		
$f_{IL}$		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) <ul style="list-style-type: none"> <li>• WTON = 0: Stops</li> <li>• WTON = 1 and WDSTBYON = 1: Oscillates</li> <li>• WTON = 1 and WDSTBYON = 0: Stops</li> </ul>		
CPU		Operation stopped		
Flash memory		Operable in low-current consumption mode		
RAM		Operation stopped. However, status before HALT mode was set is retained at voltage higher than POC detection voltage.		
Port (latch)		Status before HALT mode was set is retained		
External bus interface		Operation stopped. Status of each pin CKOUT: Continuously outputs internal system clock AD15 to AD0 or D15 to D0: High impedance A19 to A0: Retains status before HALT mode was set RD, WR0, WR1: High level ASTB: Low level WAIT: High impedance		
Timer array unit (TAU)		Operable		
Real-time counter (RTC)				
Watchdog timer		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) <ul style="list-style-type: none"> <li>• WTON = 0: Stops</li> <li>• WTON = 1 and WDSTBYON = 1: Operates</li> <li>• WTON = 1 and WDSTBYON = 0: Stops</li> </ul>		
Clock output/buzzer output		Operable		
A/D converter				
D/A converter				
Serial array unit (SAU)				
Serial interface (IIC0)				
Multiplier				
DMA controller		Operable		
Power-on-clear function				
Low-voltage detection function				
External interrupt				
Key interrupt function				

- Remarks**
1.  $f_{IH}$ : Internal high-speed oscillation clock  
 $f_x$ : X1 clock  
 $f_{EX}$ : External main system clock  
 $f_{XT}$ : XT1 clock  
 $f_{IL}$ : Internal low-speed oscillation clock
  2. Axx: Address bus  
 Dxx: Data bus  
 ADxx: Multiplexed address/data bus

Table 19-1. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting		When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock
Item		When CPU Is Operating on XT1 Clock (f <sub>XT</sub> )
System clock		Clock supply to the CPU is stopped
Main system clock	f <sub>IH</sub>	Status before HALT mode was set is retained
	f <sub>X</sub>	
	f <sub>EX</sub>	Operates or stops by external clock input
Subsystem clock	f <sub>XT</sub>	Operation continues (cannot be stopped)
f <sub>IL</sub>		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) <ul style="list-style-type: none"> <li>• WTON = 0: Stops</li> <li>• WTON = 1 and WDSTBYON = 1: Oscillates</li> <li>• WTON = 1 and WDSTBYON = 0: Stops</li> </ul>
CPU		Operation stopped
Flash memory		Operable in low-current consumption mode
RAM		Operation stopped. However, status before HALT mode was set is retained at voltage higher than POC detection voltage.
Port (latch)		Status before HALT mode was set is retained
External bus interface		Operation stopped. Status of each pin <ul style="list-style-type: none"> <li>CKOUT: Continuously outputs internal system clock</li> <li>AD15 to AD0 or D15 to D0: High impedance</li> <li>A19 to A0: Retains status before HALT mode was set</li> <li>RD, WR0, WR1: High level</li> <li>ASTB: Low level</li> <li>WAIT: High impedance</li> </ul>
Timer array unit (TAU)		Operable
Real-time counter (RTC)		
Watchdog timer		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) <ul style="list-style-type: none"> <li>• WTON = 0: Stops</li> <li>• WTON = 1 and WDSTBYON = 1: Operates</li> <li>• WTON = 1 and WDSTBYON = 0: Stops</li> </ul>
Clock output/buzzer output		Operable
A/D converter		Cannot operate
D/A converter		Operable
Serial array unit (SAU)		
Serial interface (IIC0)		Cannot operate
Multiplier		Operation stopped
DMA controller		Operable
Power-on-clear function		
Low-voltage detection function		
External interrupt		
Key interrupt function		

- Remarks**
- f<sub>IH</sub>: Internal high-speed oscillation clock

f<sub>X</sub>: X1 clock

f<sub>EX</sub>: External main system clock

f<sub>XT</sub>: XT1 clock

f<sub>IL</sub>: Internal low-speed oscillation clock
  - A<sub>xx</sub>: Address bus

D<sub>xx</sub>: Data bus

AD<sub>xx</sub>: Multiplexed address/data bus

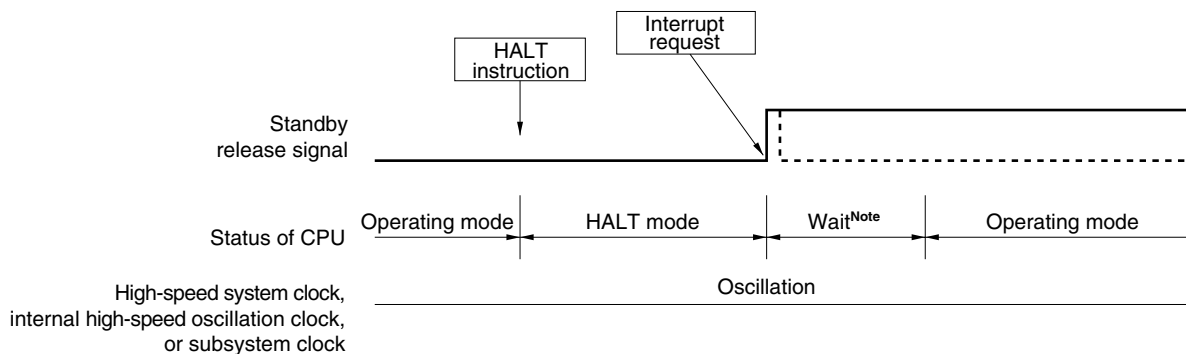
**(2) HALT mode release**

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 19-3. HALT Mode Release by Interrupt Request Generation**



**Note** The wait time is as follows:

- When vectored interrupt servicing is carried out
  - When main system clock is used: 10 to 12 clocks
  - When subsystem clock is used: 8 to 10 clocks
- When vectored interrupt servicing is not carried out
  - When main system clock is used: 5 or 6 clocks
  - When subsystem clock is used: 3 or 4 clocks

**Remark** The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

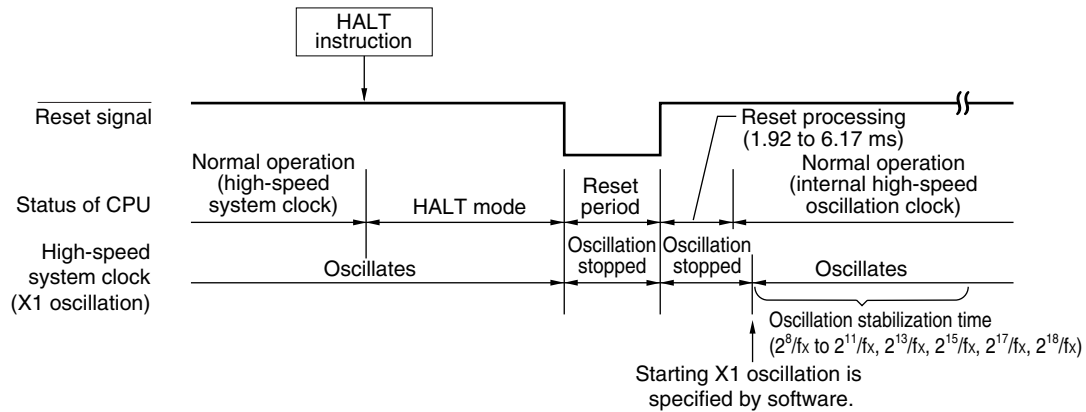


**(b) Release by reset signal generation**

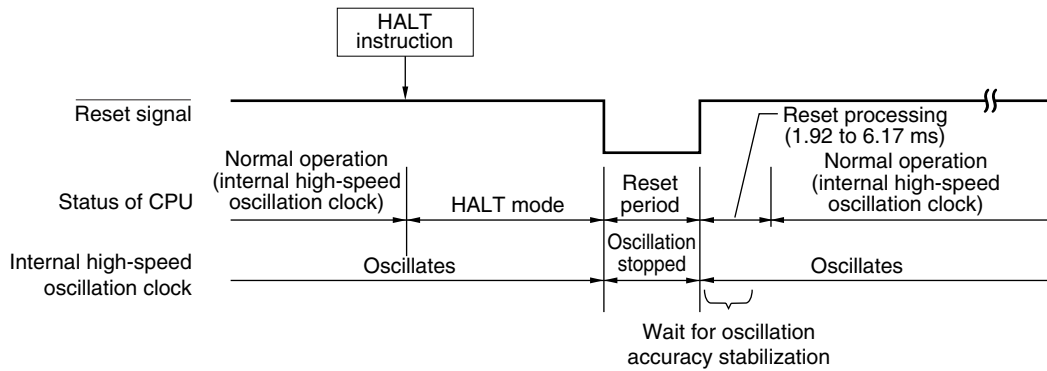
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

**Figure 19-4. HALT Mode Release by Reset**

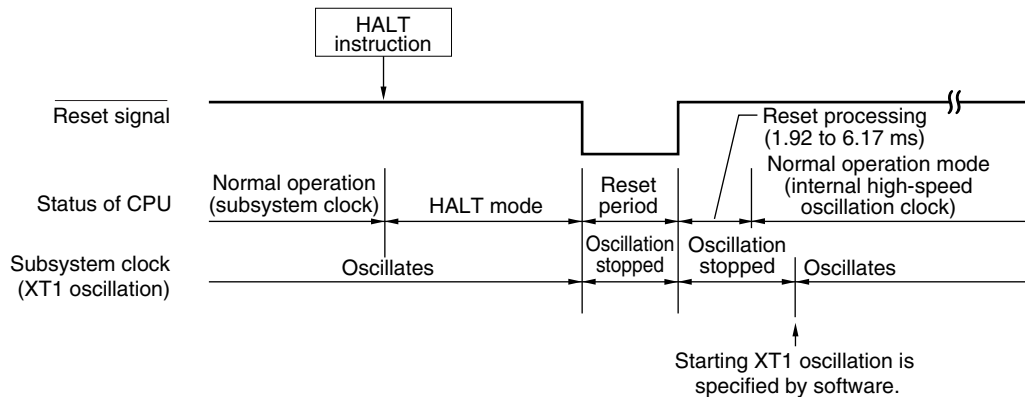
**(1) When high-speed system clock is used as CPU clock**



**(2) When internal high-speed oscillation clock is used as CPU clock**



**(3) When subsystem clock is used as CPU clock**



**Remark** fx: X1 clock oscillation frequency

## 19.2.2 STOP mode

### (1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the main system clock.

**Caution** Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

The operating statuses in the STOP mode are shown below.

Table 19-2. Operating Statuses in STOP Mode

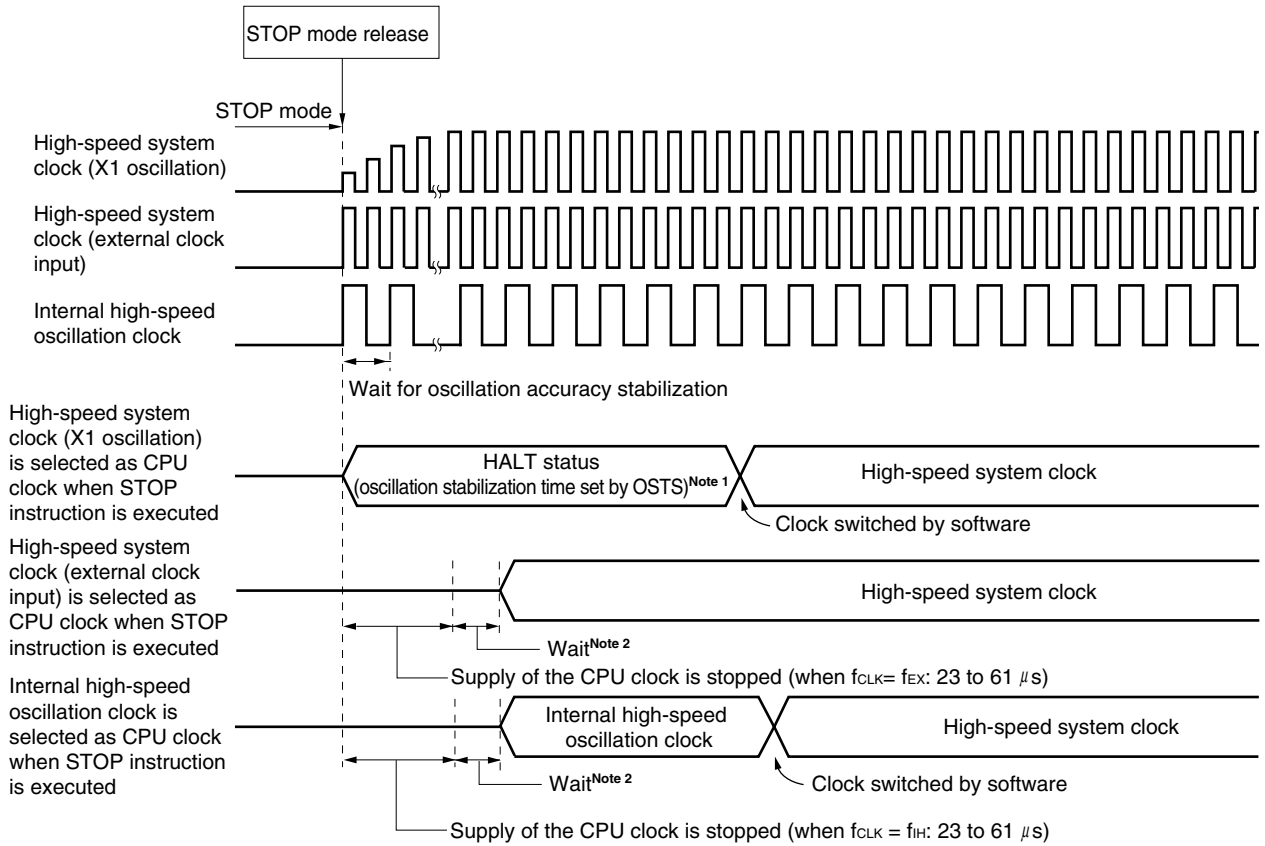
STOP Mode Setting		When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
		When CPU Is Operating on Internal High-Speed Oscillation Clock ( $f_{IH}$ )	When CPU Is Operating on X1 Clock ( $f_x$ )	When CPU Is Operating on External Main System Clock ( $f_{EX}$ )
System clock		Clock supply to the CPU is stopped		
Main system clock	$f_{IH}$	Stopped		
	$f_x$			
	$f_{EX}$			
Subsystem clock	$f_{XT}$	Status before STOP mode was set is retained		
$f_{IL}$		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) <ul style="list-style-type: none"> <li>• WTON = 0: Stops</li> <li>• WTON = 1 and WDSTBYON = 1: Oscillates</li> <li>• WTON = 1 and WDSTBYON = 0: Stops</li> </ul>		
CPU		Operation stopped		
Flash memory		Operation stopped		
RAM		Operation stopped. However, status before STOP mode was set is retained at voltage higher than POC detection voltage.		
Port (latch)		Status before STOP mode was set is retained		
External bus interface		Operation stopped. Status of each pin CKOUT: Low level AD15 to AD0 or D15 to D0: High impedance A19 to A0: Retains status before STOP mode was set RD, WR0, WR1: High level ASTB: Low level WAIT: High impedance		
Timer array unit (TAU)		Operation stopped		
Real-time counter (RTC)		Operable		
Watchdog timer		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) <ul style="list-style-type: none"> <li>• WTON = 0: Stops</li> <li>• WTON = 1 and WDSTBYON = 1: Operates</li> <li>• WTON = 1 and WDSTBYON = 0: Stops</li> </ul>		
Clock output/buzzer output		Operable only when subsystem clock is selected as the count clock		
A/D converter		Operation stopped		
D/A converter		Operation stopped (the pin in Hi-Z status)		
Serial array unit (SAU)		Operation stopped		
Serial interface (IIC0)				
Multiplier				
DMA controller				
Power-on-clear function				
Low-voltage detection function		Operable		
External interrupt				
Key interrupt function				

- Remarks**
1.  $f_{IH}$ : Internal high-speed oscillation clock
  - $f_x$ : X1 clock
  - $f_{EX}$ : External main system clock
  - $f_{XT}$ : XT1 clock
  - $f_{IL}$ : Internal low-speed oscillation clock
  2.  $A_{xx}$ : Address bus
  - $D_{xx}$ : Data bus
  - $AD_{xx}$ : Multiplexed address/data bus

- Cautions**
1. To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.
  2. To stop the internal low-speed oscillation clock in the STOP mode, use an option byte to stop the watchdog timer operation in the HALT/STOP mode (bit 0 (WDSTBYON) of 000C0H = 0), and then execute the STOP instruction.
  3. To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the internal high-speed oscillation clock before the execution of the STOP instruction. Before changing the CPU clock from the internal high-speed oscillation clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).

(2) STOP mode release

Figure 19-5. Operation Timing When STOP Mode Is Released (Release by Unmasked Interrupt Request)



**Notes** 1. When the oscillation stabilization time set by OSTs is equal to or shorter than 61  $\mu$ s, the HALT status is retained to a maximum of “61  $\mu$ s + wait time.”

2. The wait time is as follows:

- When vectored interrupt servicing is carried out: 10 to 12 clocks
- When vectored interrupt servicing is not carried out: 5 or 6 clocks

**Remark**  $f_{EX}$ : External main system clock frequency  
 $f_{IH}$ : Internal high-speed oscillation clock frequency  
 $f_{CLK}$ : CPU/peripheral hardware clock frequency

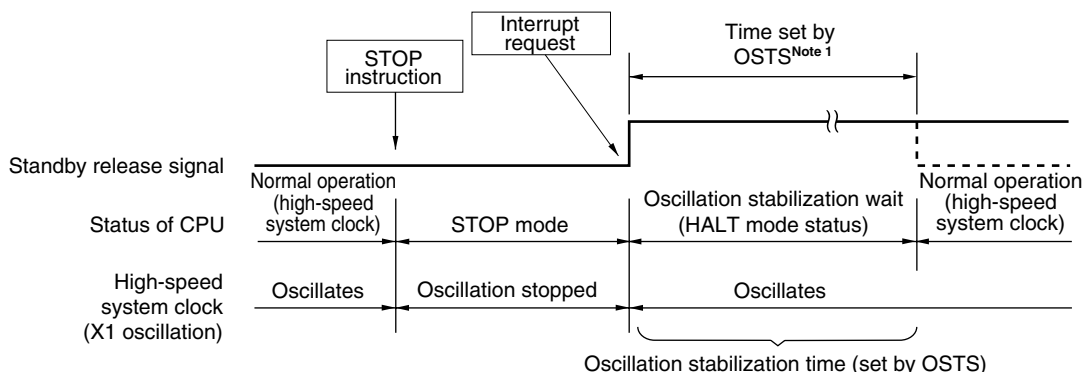
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

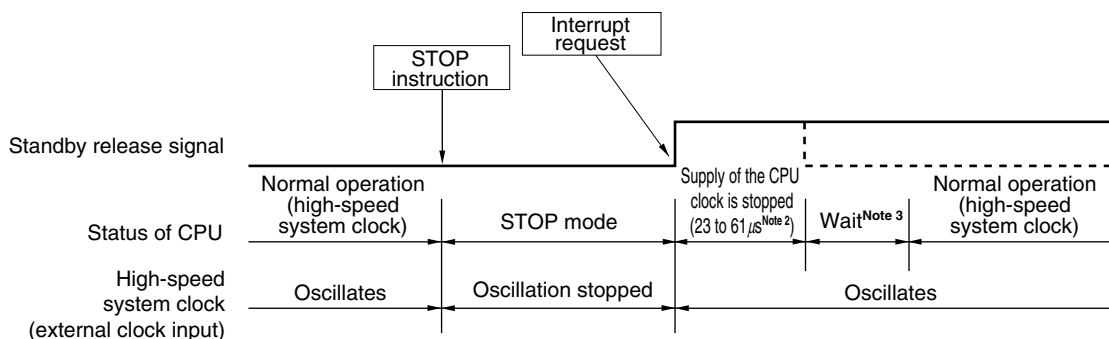
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 19-6. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed system clock (X1 oscillation) is used as CPU clock



(2) When high-speed system clock (external clock input) is used as CPU clock

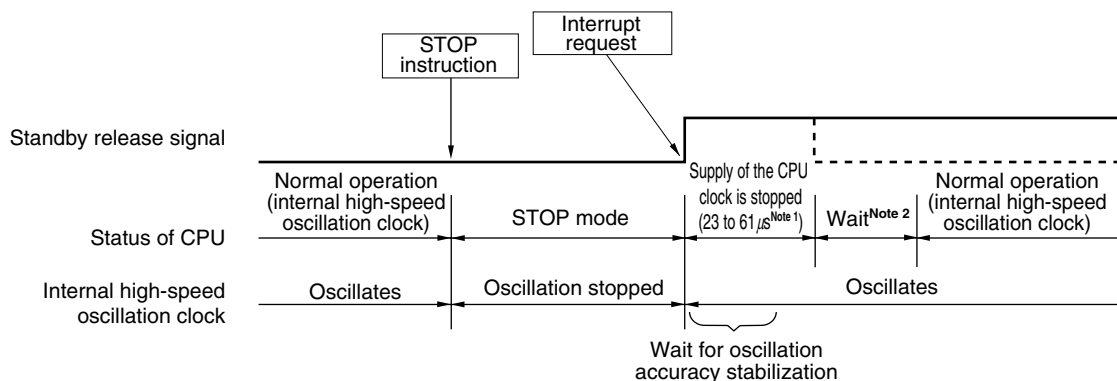


- Notes**
1. When the oscillation stabilization time set by OSTS is equal to or shorter than 61  $\mu\text{s}$ , the HALT status is retained to a maximum of "61  $\mu\text{s}$  + wait time".
  2. When  $f_{\text{CLK}} = f_{\text{EX}}$
  3. The wait time is as follows:
    - When vectored interrupt servicing is carried out: 10 to 12 clocks
    - When vectored interrupt servicing is not carried out: 5 or 6 clocks

- Remarks**
1. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.
  2.  $f_{\text{EX}}$ : External main system clock frequency  
 $f_{\text{CLK}}$ : CPU/peripheral hardware clock frequency

Figure 19-6. STOP Mode Release by Interrupt Request Generation (2/2)

## (3) When internal high-speed oscillation clock is used as CPU clock



**Notes** 1. When  $f_{CLK} = f_{IH}$

2. The wait time is as follows:

- When vectored interrupt servicing is carried out: 10 to 12 clocks
- When vectored interrupt servicing is not carried out: 5 or 6 clocks

**Remarks** 1. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

2.  $f_{IH}$ : Internal high-speed oscillation clock frequency

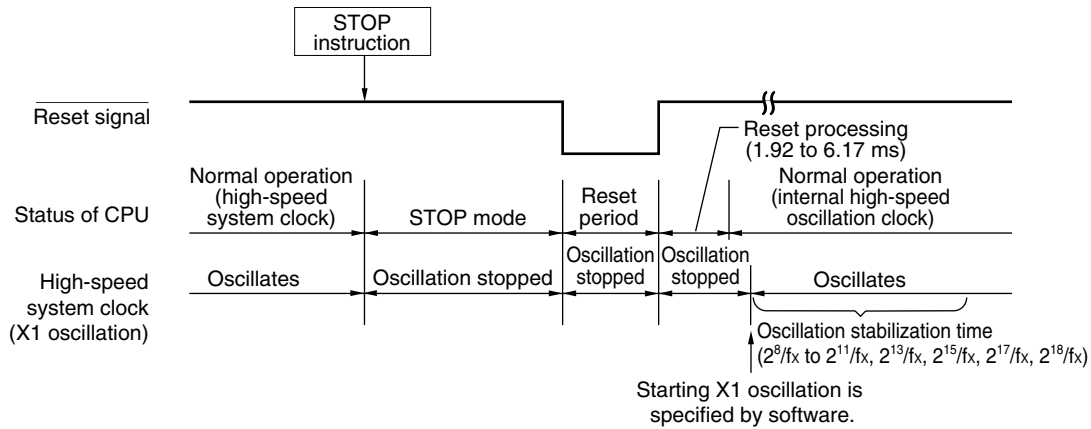
$f_{CLK}$ : CPU/peripheral hardware clock frequency

**(b) Release by reset signal generation**

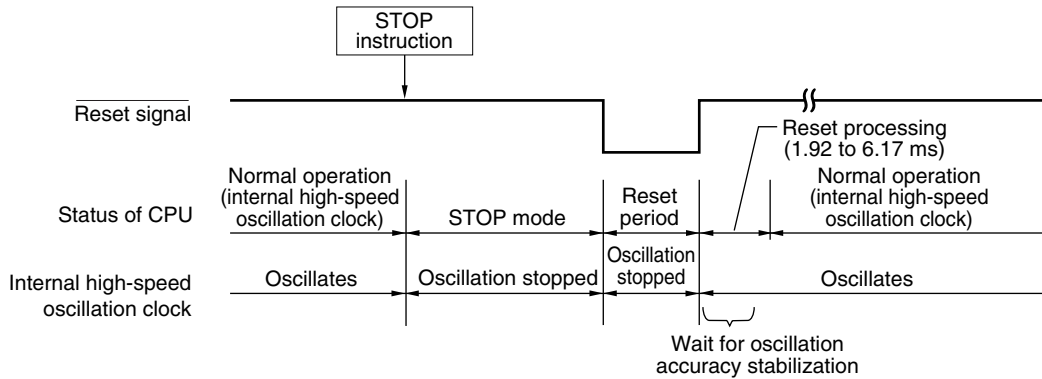
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

**Figure 19-7. STOP Mode Release by Reset**

**(1) When high-speed system clock is used as CPU clock**



**(2) When internal high-speed oscillation clock is used as CPU clock**



**Remark**  $f_x$ : X1 clock oscillation frequency



## CHAPTER 20 RESET FUNCTION

The following five operations are available to generate a reset signal.

- (1) External reset input via  $\overline{\text{RESET}}$  pin
- (2) Internal reset by watchdog timer program loop detection
- (3) Internal reset by comparison of supply voltage and detection voltage of power-on-clear (POC) circuit
- (4) Internal reset by comparison of supply voltage of the low-voltage detector (LVI) or input voltage (EXLVI) from external input pin, and detection voltage
- (5) Internal reset by execution of illegal instruction<sup>Note</sup>

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.

A reset is effected when a low level is input to the  $\overline{\text{RESET}}$  pin, the watchdog timer overflows, or by POC and LVI circuit voltage detection or execution of illegal instruction<sup>Note</sup>, and each item of hardware is set to the status shown in Tables 20-1 and 20-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release, except for P130, which is low-level output.

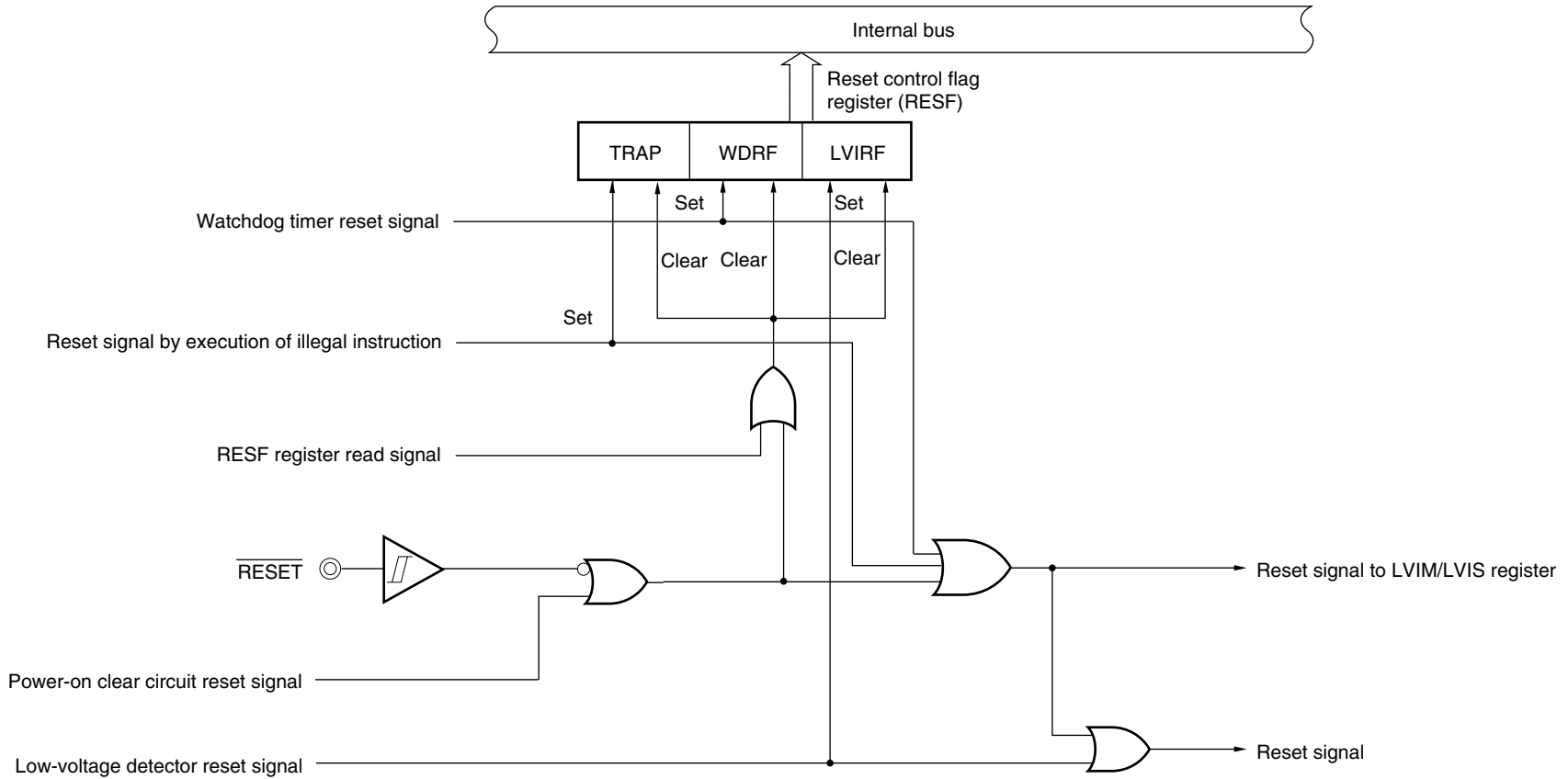
When a low level is input to the  $\overline{\text{RESET}}$  pin, the device is reset. It is released from the reset status when a high level is input to the  $\overline{\text{RESET}}$  pin and program execution is started with the internal high-speed oscillation clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the internal high-speed oscillation clock (see **Figures 20-2 to 20-4**) after reset processing. Reset by POC and LVI circuit supply voltage detection is automatically released when  $V_{DD} \geq V_{POC}$  or  $V_{DD} \geq V_{LVI}$  after the reset, and program execution starts using the internal high-speed oscillation clock (see **CHAPTER 21 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 22 LOW-VOLTAGE DETECTOR**) after reset processing.

**Note** The illegal instruction is generated when instruction code FFH is executed.

Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

- Cautions**
1. For an external reset, input a low level for 10  $\mu\text{s}$  or more to the  $\overline{\text{RESET}}$  pin.  
(If an external reset is effected upon power application, the period during which the supply voltage is outside the operating range ( $V_{DD} < 1.8 \text{ V}$ ) is not counted in the 10  $\mu\text{s}$ . However, the low-level input may be continued before POC is released.)
  2. During reset input, the X1 clock, XT1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.
  3. When the STOP mode is released by a reset, the RAM contents in the STOP mode are held during reset input. However, because SFR and 2nd SFR are initialized, the port pins become high-impedance, except for P130, which is set to low-level output.

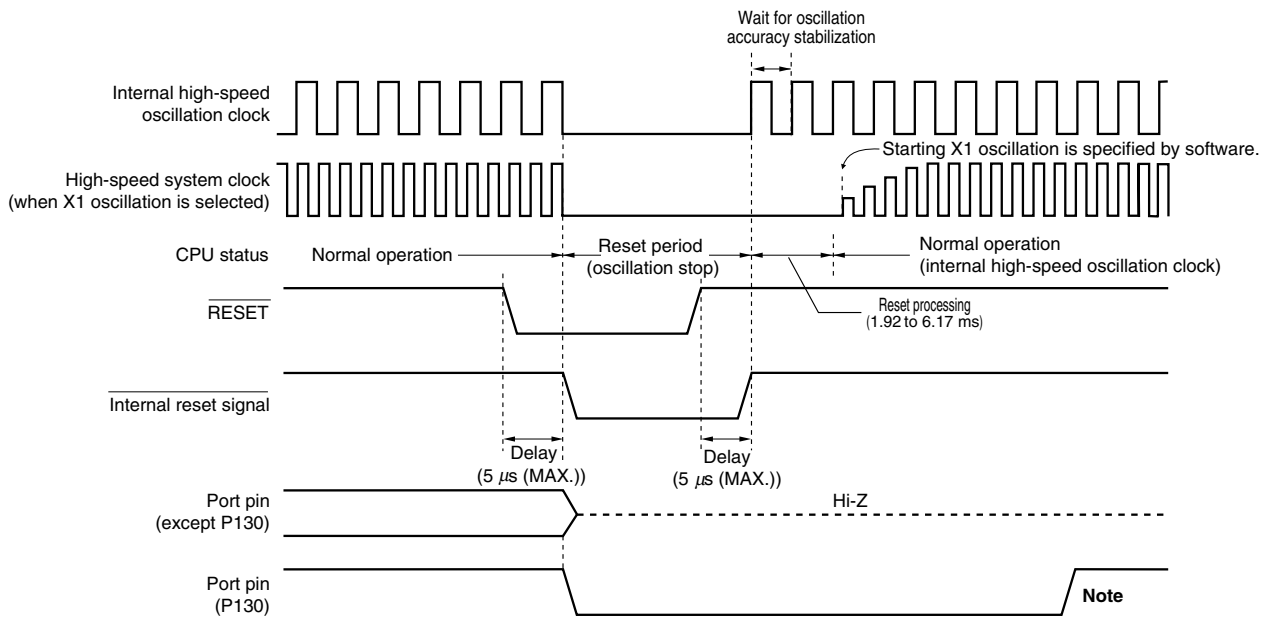
Figure 20-1. Block Diagram of Reset Function



**Caution** An LVI circuit internal reset does not reset the LVI circuit.

- Remarks**
1. LVIM: Low-voltage detection register
  2. LVIS: Low-voltage detection level select register

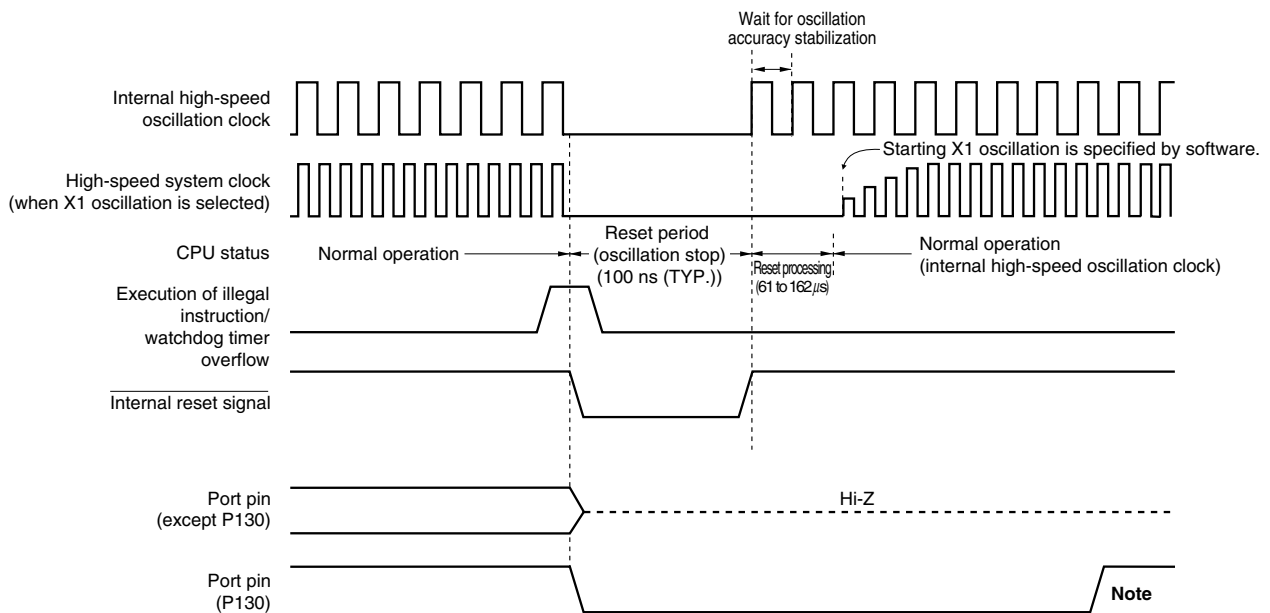
Figure 20-2. Timing of Reset by RESET Input



**Note** Set P130 to high-level output by software.

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

Figure 20-3. Timing of Reset Due to Execution of Illegal Instruction or Watchdog Timer Overflow

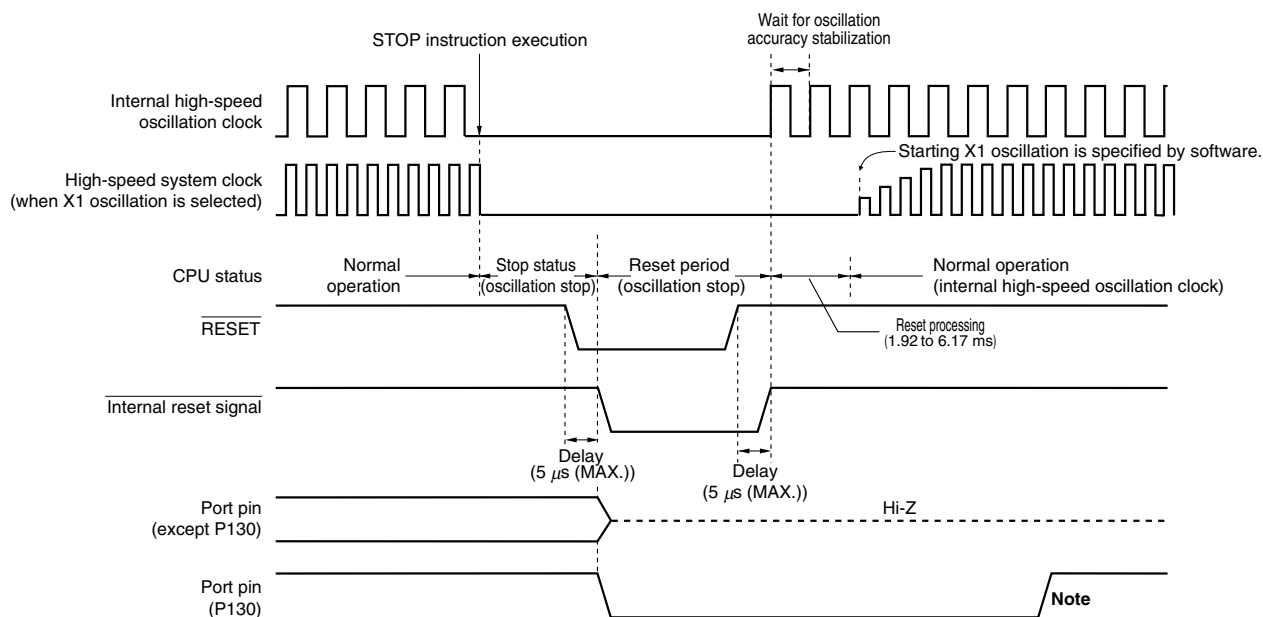


**Note** Set P130 to high-level output by software.

**Caution** A watchdog timer internal reset resets the watchdog timer.

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

Figure 20-4. Timing of Reset in STOP Mode by  $\overline{\text{RESET}}$  Input



**Note** Set P130 to high-level output by software.

- Remarks**
1. When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.
  2. For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 21 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 22 LOW-VOLTAGE DETECTOR**.

Table 20-1. Operation Statuses During Reset Period

Item	During Reset Period			
System clock	Clock supply to the CPU is stopped.			
Main system clock	f <sub>IH</sub>	Operation stopped		
	f <sub>X</sub>	Operation stopped (X1 and X2 pins are input port mode)		
	f <sub>EX</sub>	Clock input invalid (pin is input port mode)		
Subsystem clock	f <sub>XT</sub>	Operation stopped (XT1 and XT2 pins are input port mode)		
	f <sub>IL</sub>	Operation stopped		
CPU				
Flash memory	Operable in low-current consumption mode			
RAM	Operation stopped			
Port (latch)	Operation stopped			
External bus interface				
Timer array unit (TAU)				
Real-time counter (RTC)				
Watchdog timer				
Clock output/buzzer output				
A/D converter				
D/A converter				
Serial array unit (SAU)				
Serial interface (IIC0)				
Multiplier				
DMA controller				
Power-on-clear function			Operable	
Low-voltage detection function			Operation stopped (however, operation continues at LVI reset)	
External interrupt	Operation stopped			
Key interrupt function				

**Remark** f<sub>IH</sub>: Internal high-speed oscillation clock  
f<sub>X</sub>: X1 oscillation clock  
f<sub>EX</sub>: External main system clock  
f<sub>XT</sub>: XT1 oscillation clock  
f<sub>IL</sub>: Internal low-speed oscillation clock

Table 20-2. Hardware Statuses After Reset Acknowledgment (1/3)

Hardware		After Reset Acknowledgment <sup>Note 1</sup>
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 <sup>Note 2</sup>
	General-purpose registers	Undefined <sup>Note 2</sup>
Port registers (P0 to P9, P11 to P16) (output latches)		00H
Port mode registers (PM0 to PM9, PM11 to PM16)		FFH
Port input mode registers 0, 4, 9, 12, 14 (PIM0, PIM4, PIM9, PIM12, PIM14)		00H
Port output mode registers 0, 4, 9, 12, 14 (POM0, POM4, POM9, POM12, POM14)		00H
Pull-up resistor option registers (PU0, PU1, PU3 to PU9, PU11 to PU14, PU16)		00H
Memory extension mode control register (MEM)		00H
Clock operation mode control register (CMC)		00H
Clock operation status control register (CSC)		C0H
Processor mode control register (PMC)		00H
System clock control register (CKC)		09H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		07H
Noise filter enable registers 0, 1, 2 (NFEN0, NFEN1, NFEN2)		00H
Peripheral enable registers 0, 1 (PER0, PER1)		00H
Internal high-speed oscillator trimming register (HIOTRM)		10H
Operation speed mode control register (OSMC)		00H
Timer array unit (TAU)	Timer data registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13 (TDR00, TDR01, TDR02, TDR03, TDR04, TDR05, TDR06, TDR07, TDR10, TDR11, TDR12, TDR13)	0000H
	Timer mode registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13 (TMR00, TMR01, TMR02, TMR03, TMR04, TMR05, TMR06, TMR07, TMR10, TMR11, TMR12, TMR13)	0000H
	Timer status registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13 (TSR00, TSR01, TSR02, TSR03, TSR04, TSR05, TSR06, TSR07, TSR10, TSR11, TSR12, TSR13)	0000H
	Timer input select registers 0, 1 (TIS0, TIS1)	00H
	Timer counter registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13 (TCR00, TCR01, TCR02, TCR03, TCR04, TCR05, TCR06, TCR07, TCR10, TCR11, TCR12, TCR13)	FFFFH
	Timer channel enable status registers 0, 1 (TE0, TE1)	0000H
	Timer channel start registers 0, 1 (TS0, TS1)	0000H
	Timer channel stop registers 0, 1 (TT0, TT1)	0000H
	Timer clock select registers 0, 1 (TPS0, TPS1)	0000H
	Timer output registers 0, 1 (TO0, TO1)	0000H
	Timer output enable registers 0, 1 (TOE0, TOE1)	0000H
	Timer output level registers 0, 1 (TOL0, TOL1)	0000H
Timer output mode registers 0, 1 (TOM0, TOM1)	0000H	

**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. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

Table 20-2. Hardware Statuses After Reset Acknowledgment (2/3)

Hardware		Status After Reset Acknowledgment <sup>Note 1</sup>
Real-time counter	Sub-count register (RSUBC)	0000H
	Second count register (SEC)	00H
	Minute count register (MIN)	00H
	Hour count register (HOUR)	12H
	Day count register (DAY)	01H
	Week count register (WEEK)	00H
	Month count register (MONTH)	01H
	Year count register (YEAR)	00H
	Watch error correction register (SUBCUD)	00H
	Alarm minute register (ALARMWWM)	00H
	Alarm hour register (ALARMWH)	12H
	Alarm week register (ALARMWW)	00H
	Real-time counter control register 0 (RTCC0)	00H
	Real-time counter control register 1 (RTCC1)	00H
	Real-time counter control register 2 (RTCC2)	00H
Clock output/buzzer output controller	Clock output select registers 0, 1 (CKS0, CKS1)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH <sup>Note 2</sup>
A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	Mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	10H
D/A converter	8-bit D/A conversion value setting registers 0, 1 (DACS0, DACS1)	00H
	D/A converter mode register (DAM)	00H
Serial array unit (SAU)	Serial data registers 00, 01, 02, 03, 10, 11, 12, 13 (SDR00, SDR01, SDR02, SDR03, SDR10, SDR11, SDR12, SDR13)	0000H
	Serial status registers 00, 01, 02, 03, 10, 11, 12, 13 (SSR00, SSR01, SSR02, SSR03, SSR10, SSR11, SSR12, SSR13)	0000H
	Serial flag clear trigger registers 00, 01, 02, 03, 10, 11, 12, 13 (SIR00, SIR01, SIR02, SIR03, SIR10, SIR11, SIR12, SIR13)	0000H
	Serial mode registers 00, 01, 02, 03, 10, 11, 12, 13 (SMR00, SMR01, SMR02, SMR03, SMR10, SMR11, SMR12, SMR13)	0020H
	Serial communication operation setting registers 00, 01, 02, 03, 10, 11, 12, 13 (SCR00, SCR01, SCR02, SCR03, SCR10, SCR11, SCR12, SCR13)	0087H
	Serial channel enable status registers 0, 1 (SE0, SE1)	0000H
	Serial channel start registers 0, 1 (SS0, SS1)	0000H
	Serial channel stop registers 0, 1 (ST0, ST1)	0000H
	Serial clock select registers 0, 1 (SPS0, SPS1)	0000H
	Serial output registers 0, 1 (SO0, SO1)	0F0FH
	Serial output enable registers 0, 1 (SOE0, SOE1)	0000H
	Serial output level registers 0, 1 (SOL0, SOL1)	0000H
	Input switch control register (ISC)	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. The reset value of WDTE is determined by the option byte setting.

**Table 20-2. Hardware Statuses After Reset Acknowledgment (3/3)**

Hardware		Status After Reset Acknowledgment <sup>Note 1</sup>
Serial interface IIC0	Shift register 0 (IIC0)	00H
	Control register 0 (IICC0)	00H
	Slave address register 0 (SVA0)	00H
	Clock select register 0 (IICCL0)	00H
	Function expansion register 0 (IICX0)	00H
	Status register 0 (IICS0)	00H
	Flag register 0 (IICF0)	00H
Multiplier	Multiplication input data register A (MULA)	0000H
	Multiplication input data register B (MULB)	0000H
	Higher multiplication result storage register (MULOH)	0000H
	Lower multiplication result storage register (MULOL)	0000H
Key interrupt	Key return mode register (KRM)	00H
Reset function	Reset control flag register (RESF)	00H <sup>Note 2</sup>
Low-voltage detector	Low-voltage detection register (LVIM)	00H <sup>Note 3</sup>
	Low-voltage detection level select register (LVIS)	0EH <sup>Note 2</sup>
Regulator	Regulator mode control register (RMC)	00H
DMA controller	SFR address registers 0, 1 (DSA0, DSA1)	00H
	RAM address registers 0L, 0H, 1L, 1H (DRA0L, DRA0H, DRA1L, DRA1H)	00H
	Byte count registers 0L, 0H, 1L, 1H (DBC0L, DBC0H, DBC1L, DBC1H)	00H
	Mode control registers 0, 1 (DMC0, DMC1)	00H
	Operation control registers 0, 1 (DRC0, DRC1)	00H
Interrupt	Request flag registers 0L, 0H, 1L, 1H, 2L, 2H (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)	00H
	Mask flag registers 0L, 0H, 1L, 1H, 2L, 2H (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)	FFH
	Priority specification flag registers 00L, 00H, 01L, 01H, 02L, 02H, 10L, 10H, 11L, 11H, 12L, 12H (PR00L, PR00H, PR01L, PR01H, PR10L, PR10H, PR11L, PR11H, PR02L, PR02H, PR12L, PR12H)	FFH
	External interrupt rising edge enable registers 0, 1 (EGP0, EGP1)	00H
	External interrupt falling edge enable registers 0, 1 (EGN0, EGN1)	00H
BCD correction circuit	BCD correction result register (BCDADJ)	Undefined

**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		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by LVI
RESF	TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held
	WDRF bit			Held	Set (1)	Held
	LVIRF bit			Held	Held	Set (1)
LVIS		Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Held

**3.** This value varies depending on the reset source and the option byte.



## 20.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0R/KH3. The reset control flag register (RESF) is used to store which source has generated the reset request.

RESF can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$  input, reset by power-on-clear (POC) circuit, and reading RESF set RESF to 00H.

**Figure 20-5. Format of Reset Control Flag Register (RESF)**

Address: FFFA8H After reset: 00H<sup>Note 1</sup> R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDRF	0	0	0	LVIRF

TRAP	Internal reset request by execution of illegal instruction <sup>Note 2</sup>
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

WDRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by low-voltage detector (LVI)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

**Notes 1.** The value after reset varies depending on the reset source.

**2.** When instruction code FFH is executed.

Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

**Cautions 1.** Do not read data by a 1-bit memory manipulation instruction.

**2.** When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, LVIRF flag may become 1 from the beginning depending on the power-on waveform.

The status of RESF when a reset request is generated is shown in Table 20-3.

**Table 20-3. RESF Status When Reset Request Is Generated**

Reset Source / Flag	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by LVI
TRAP	Cleared (0)	Cleared (0)	Set (1)	Held	Held
WDRF			Held	Set (1)	Held
LVIRF			Held	Held	Set (1)

## CHAPTER 21 POWER-ON-CLEAR CIRCUIT

### 21.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) has the following functions.

- Generates internal reset signal at power on.  
The reset signal is released when the supply voltage ( $V_{DD}$ ) exceeds  $1.59\text{ V} \pm 0.09\text{ V}$ .

**Caution** If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage ( $V_{DD}$ ) exceeds  $2.07\text{ V} \pm 0.2\text{ V}$ .

- Compares supply voltage ( $V_{DD}$ ) and detection voltage ( $V_{POC} = 1.59\text{ V} \pm 0.09\text{ V}$ ), generates internal reset signal when  $V_{DD} < V_{POC}$ .

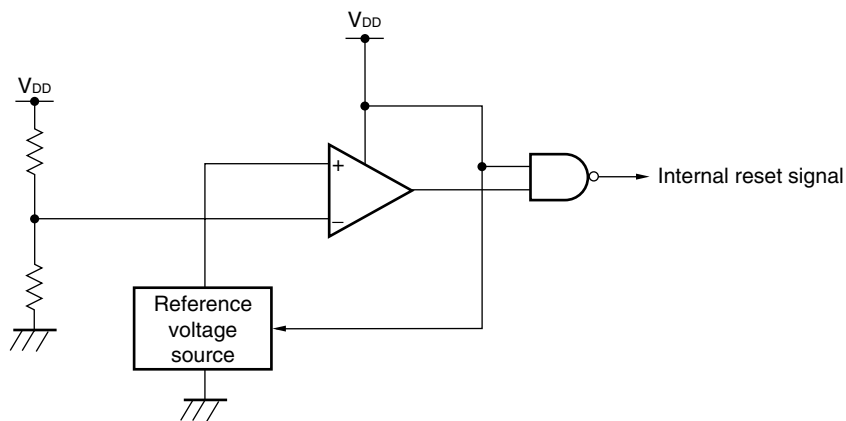
**Caution** If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

**Remark** This product incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), low-voltage detector (LVI), or illegal instruction execution. RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT, LVI or illegal instruction execution. For details of RESF, see **CHAPTER 20 RESET FUNCTION**.

## 21.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 21-1.

Figure 21-1. Block Diagram of Power-on-Clear Circuit



## 21.3 Operation of Power-on-Clear Circuit

- An internal reset signal is generated on power application. When the supply voltage ( $V_{DD}$ ) exceeds the detection voltage ( $V_{POC} = 1.59\text{ V} \pm 0.09\text{ V}$ ), the reset status is released.

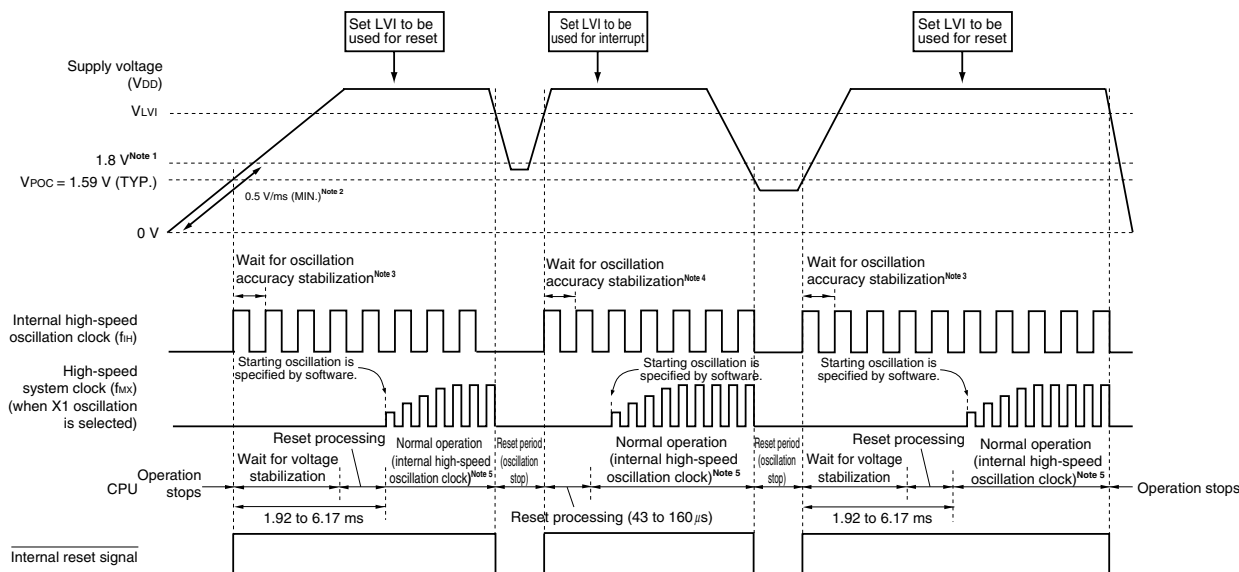
**Caution** If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage ( $V_{DD}$ ) exceeds  $2.07\text{ V} \pm 0.2\text{ V}$ .

- The supply voltage ( $V_{DD}$ ) and detection voltage ( $V_{POC} = 1.59\text{ V} \pm 0.09\text{ V}$ ) are compared. When  $V_{DD} < V_{POC}$ , the internal reset signal is generated.

The timing of generation of the internal reset signal by the power-on-clear circuit and low-voltage detector is shown below.

Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)

(1) When LVI is OFF upon power application (option byte: LVIOFF = 1)



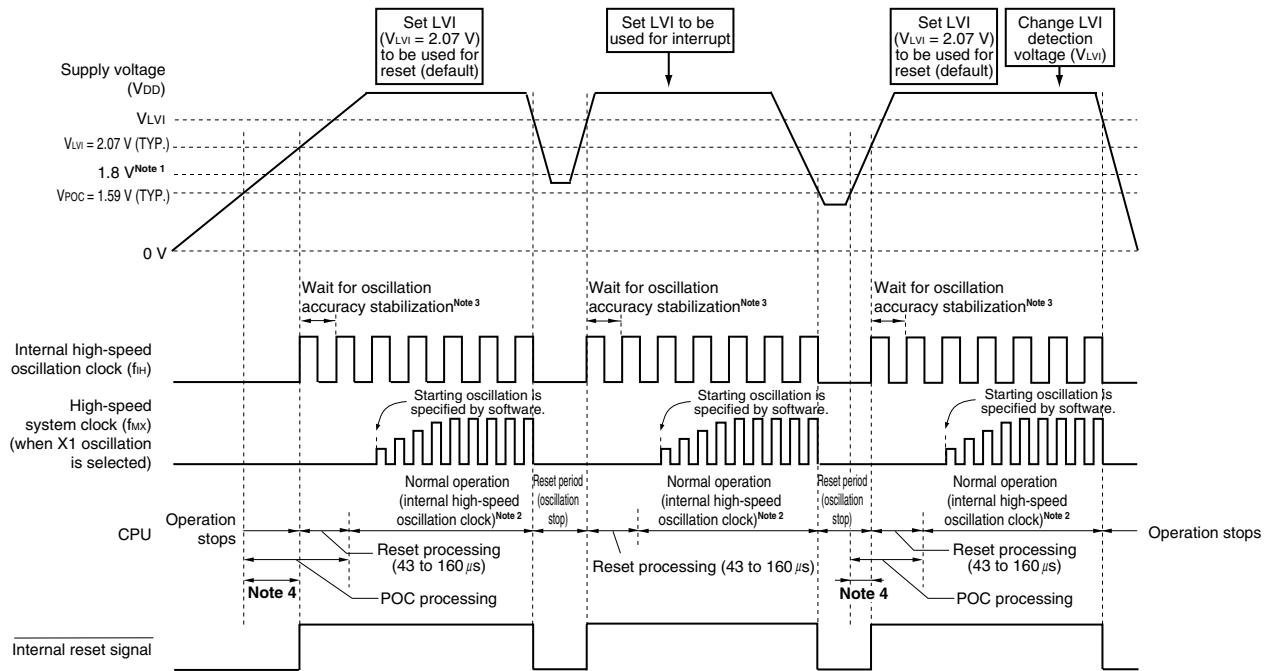
- Notes**
1. The operation guaranteed range is  $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ . To make the state at lower than 1.8 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the  $\overline{\text{RESET}}$  pin.
  2. If the rate at which the voltage rises to 1.8 V after power application is slower than 0.5 V/ms (MIN.), input a low level to the  $\overline{\text{RESET}}$  pin before the voltage reaches to 1.8 V, or set LVI to ON by default by using an option byte (option byte: LVIOFF = 0).
  3. The internal voltage stabilization time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.
  4. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.
  5. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

**Caution** Set the low-voltage detector by software after the reset status is released (see CHAPTER 22 LOW-VOLTAGE DETECTOR).

**Remark** V<sub>LVI</sub>: LVI detection voltage  
V<sub>POC</sub>: POC detection voltage

Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)

(2) When LVI is ON upon power application (option byte: LVIOFF = 0)



- Notes**
1. The operation guaranteed range is  $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ . To make the state at lower than 1.8 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the  $\overline{\text{RESET}}$  pin.
  2. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.
  3. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.
  4. The following times are required between reaching the POC detection voltage (1.59 V (TYP.)) and starting normal operation.
    - When the time to reach 2.07 V (TYP.) from 1.59 V (TYP.) is less than 6.17 ms:  
A POC processing time of 1.92 to 6.33 ms is required between reaching 1.59 V (TYP.) and starting normal operation.
    - When the time to reach 2.07 V (TYP.) from 1.59 V (TYP.) is greater than 6.17 ms:  
A reset processing time of 43 to 160  $\mu\text{s}$  is required between reaching 2.07 V (TYP.) and starting normal operation.

**Caution** Set the low-voltage detector by software after the reset status is released (see CHAPTER 22 LOW-VOLTAGE DETECTOR).

**Remark**  $V_{LVI}$ : LVI detection voltage  
 $V_{POC}$ : POC detection voltage

### 21.4 Cautions for Power-on-Clear Circuit

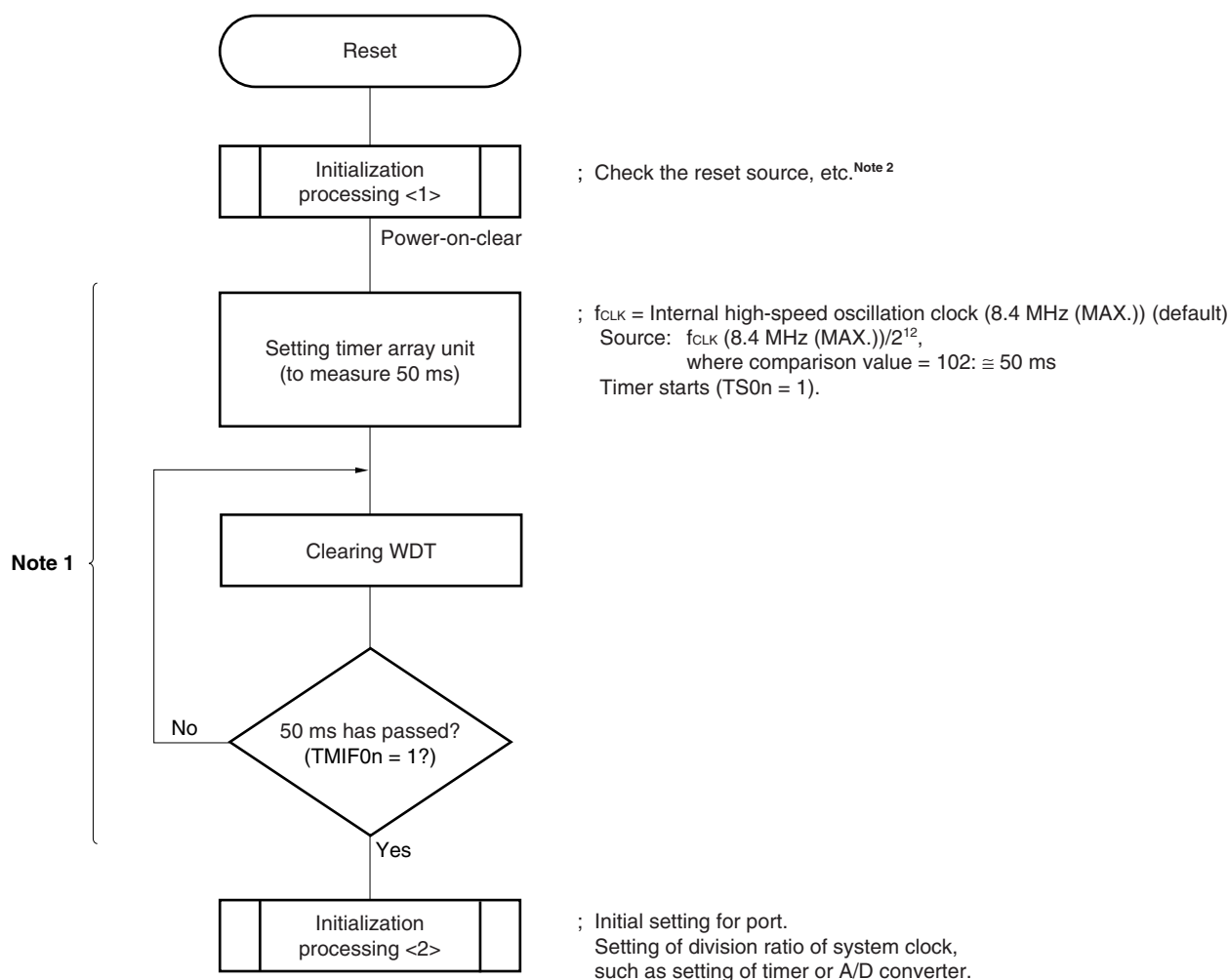
In a system where the supply voltage ( $V_{DD}$ ) fluctuates for a certain period in the vicinity of the POC detection voltage ( $V_{POC}$ ), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

**Figure 21-3. Example of Software Processing After Reset Release (1/2)**

- If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage

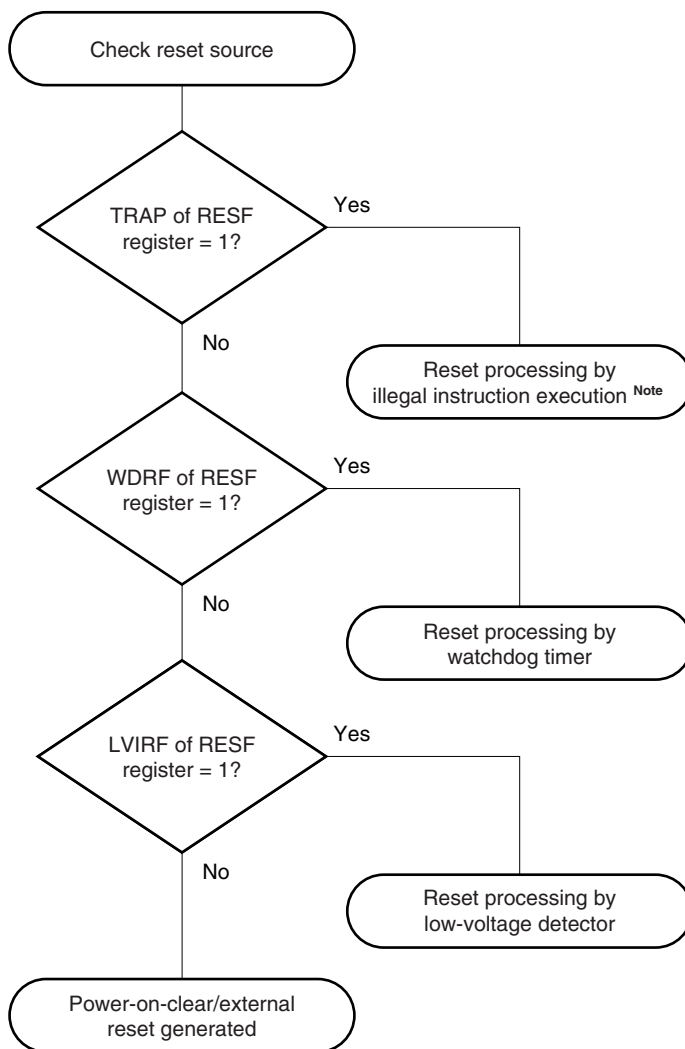


- Notes**
1. If reset is generated again during this period, initialization processing <2> is not started.
  2. A flowchart is shown on the next page.

**Remark** n: Channel number (n = 0 to 7)

Figure 21-3. Example of Software Processing After Reset Release (2/2)

- Checking reset source



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

## CHAPTER 22 LOW-VOLTAGE DETECTOR

### 22.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) has the following functions.

- The LVI circuit compares the supply voltage ( $V_{DD}$ ) with the detection voltage ( $V_{LVI}$ ) or the input voltage from an external input pin (EXLVI) with the detection voltage ( $V_{EXLVI} = 1.21\text{ V} \pm 0.1\text{ V}$ ), and generates an internal reset<sup>Note</sup> or internal interrupt signal.
- The low-voltage detector (LVI) can be set to ON by an option byte by default. If it is set to ON to raise the power supply from the POC detection voltage or lower, the internal reset signal<sup>Note</sup> is generated when the supply voltage ( $V_{DD}$ ) < detection voltage ( $V_{LVI} = 2.07\text{ V} \pm 0.2\text{ V}$ ). After that, the internal reset signal<sup>Note</sup> is generated when the supply voltage ( $V_{DD}$ ) < detection voltage ( $V_{LVI} = 2.07\text{ V} \pm 0.1\text{ V}$ ).
- The supply voltage ( $V_{DD}$ ) or the input voltage from the external input pin (EXLVI) can be selected to be detected by software.
- A reset or an interrupt can be selected to be generated after detection by software.
- Detection levels ( $V_{LVI}$ , 16 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

**Note** See the timing in **Figure 21-2 (2) When LVI is ON upon power application (option byte: LVIOFF = 0)** for the reset processing time until the normal operation is entered after the LVI reset is released.

The reset and interrupt signals are generated as follows depending on selection by software.

Selection of Level Detection of Supply Voltage ( $V_{DD}$ ) (LVISEL = 0)		Selection Level Detection of Input Voltage from External Input Pin (EXLVI) (LVISEL = 1)	
Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).	Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).
Generates an internal reset signal when $V_{DD} < V_{LVI}$ and releases the reset signal when $V_{DD} \geq V_{LVI}$ .	Generates an internal interrupt signal when $V_{DD}$ drops lower than $V_{LVI}$ ( $V_{DD} < V_{LVI}$ ) or when $V_{DD}$ becomes $V_{LVI}$ or higher ( $V_{DD} \geq V_{LVI}$ ).	Generates an internal reset signal when $EXLVI < V_{EXLVI}$ and releases the reset signal when $EXLVI \geq V_{EXLVI}$ .	Generates an internal interrupt signal when EXLVI drops lower than $V_{EXLVI}$ ( $EXLVI < V_{EXLVI}$ ) or when EXLVI becomes $V_{EXLVI}$ or higher ( $EXLVI \geq V_{EXLVI}$ ).

**Remark** LVISEL: Bit 2 of low-voltage detection register (LVIM)  
LVIMD: Bit 1 of LVIM

While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

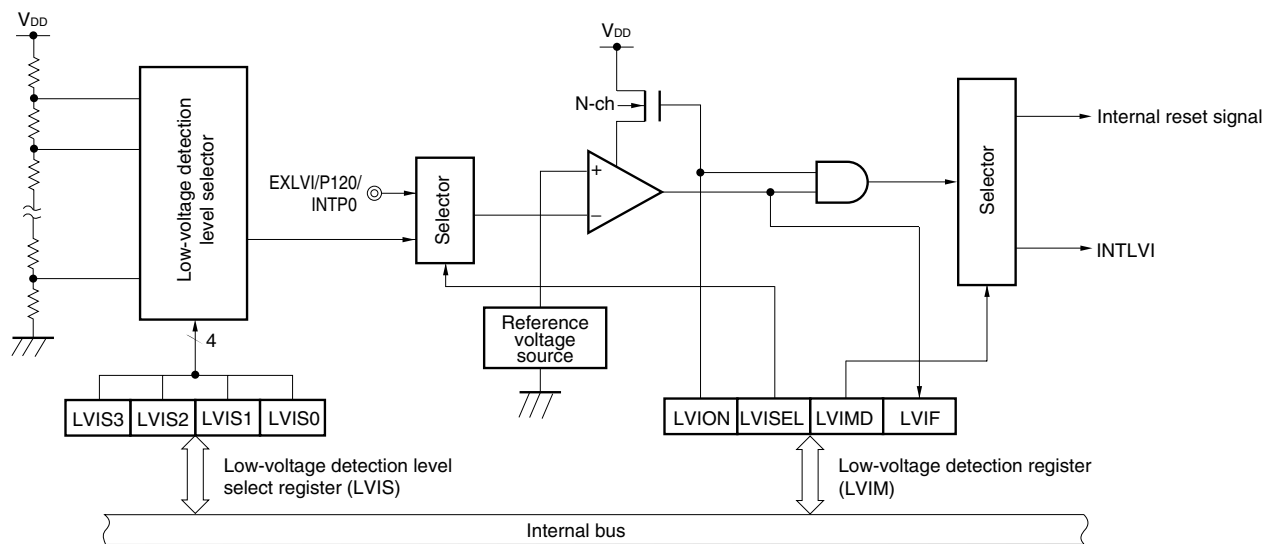
When the low-voltage detector is used to reset, bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of RESF, see **CHAPTER 20 RESET FUNCTION**.



## 22.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 22-1.

Figure 22-1. Block Diagram of Low-Voltage Detector



## 22.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection register (LVIM)
- Low-voltage detection level select register (LVIS)
- Port mode register 12 (PM12)

### (1) Low-voltage detection register (LVIM)

This register sets low-voltage detection and the operation mode.

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

Reset signal generation clears this register to 00H.

**Figure 22-2. Format of Low-Voltage Detection Register (LVIM)**

Address: FFFA9H After reset: 00H<sup>Note 1</sup> R/W<sup>Note 2</sup>

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
LVIM	LVION	0	0	0	0	LVISEL	LVIMD	LVIF

LVION <sup>Notes 3, 4</sup>	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVISEL <sup>Note 3</sup>	Voltage detection selection
0	Detects level of supply voltage ( $V_{DD}$ )
1	Detects level of input voltage from external input pin (EXLVI)

LVIMD <sup>Note 3</sup>	Low-voltage detection operation mode (interrupt/reset) selection
0	<ul style="list-style-type: none"> <li>LVISEL = 0: Generates an internal interrupt signal when the supply voltage (<math>V_{DD}</math>) drops lower than the detection voltage (<math>V_{LVI}</math>) (<math>V_{DD} &lt; V_{LVI}</math>) or when <math>V_{DD}</math> becomes <math>V_{LVI}</math> or higher (<math>V_{DD} \geq V_{LVI}</math>).</li> <li>LVISEL = 1: Generates an interrupt signal when the input voltage from an external input pin (EXLVI) drops lower than the detection voltage (<math>V_{EXLVI}</math>) (<math>EXLVI &lt; V_{EXLVI}</math>) or when EXLVI becomes <math>V_{EXLVI}</math> or higher (<math>EXLVI \geq V_{EXLVI}</math>).</li> </ul>
1	<ul style="list-style-type: none"> <li>LVISEL = 0: Generates an internal reset signal when the supply voltage (<math>V_{DD}</math>) &lt; detection voltage (<math>V_{LVI}</math>) and releases the reset signal when <math>V_{DD} \geq V_{LVI}</math>.</li> <li>LVISEL = 1: Generates an internal reset signal when the input voltage from an external input pin (EXLVI) &lt; detection voltage (<math>V_{EXLVI}</math>) and releases the reset signal when <math>EXLVI \geq V_{EXLVI}</math>.</li> </ul>

LVIF	Low-voltage detection flag
0	<ul style="list-style-type: none"> <li>LVISEL = 0: Supply voltage (<math>V_{DD}</math>) <math>\geq</math> detection voltage (<math>V_{LVI}</math>), or when LVI operation is disabled</li> <li>LVISEL = 1: Input voltage from external input pin (EXLVI) <math>\geq</math> detection voltage (<math>V_{EXLVI}</math>), or when LVI operation is disabled</li> </ul>
1	<ul style="list-style-type: none"> <li>LVISEL = 0: Supply voltage (<math>V_{DD}</math>) &lt; detection voltage (<math>V_{LVI}</math>)</li> <li>LVISEL = 1: Input voltage from external input pin (EXLVI) &lt; detection voltage (<math>V_{EXLVI}</math>)</li> </ul>

- Notes**
- The reset value changes depending on the reset source and the setting of the option byte. This register is not cleared (00H) by LVI reset. It is set to “82H” when a reset signal other than LVI is applied if option byte LVIOFF = 0, and to “00H” if option byte LVIOFF = 1.
  - Bit 0 is read-only.
  - LVION, LVIMD, and LVISEL are cleared to 0 in the case of a reset other than an LVI reset. These are not cleared to 0 in the case of an LVI reset.

**Note 4.** When LVION is set to 1, operation of the comparator in the LVI circuit is started. Use software to wait for the following periods of time, between when LVION is set to 1 and when the voltage is confirmed with LVIF.

- Operation stabilization time (10  $\mu$ S (MAX.))
- Minimum pulse width (200  $\mu$ S (MIN.))
- Detection delay time (200  $\mu$ S (MAX.))

The LVIF value for these periods may be set/cleared regardless of the voltage level, and can therefore not be used. Also, the LVIIIF interrupt request flag may be set to 1 in these periods.

**Cautions 1.** To stop LVI, follow either of the procedures below.

- When using 8-bit memory manipulation instruction: Write 00H to LVIM.
  - When using 1-bit memory manipulation instruction: Clear LVION to 0.
2. Input voltage from external input pin (EXLVI) must be  $EXLVI < V_{DD}$ .
  3. When LVI is used in interrupt mode (LVIMD = 0) and LVISEL is set to 0, an interrupt request signal (INTLVI) that disables LVI operation (clears LVION) when the supply voltage ( $V_{DD}$ ) is less than or equal to the detection voltage ( $V_{LVI}$ ) (if LVISEL = 1, input voltage of external input pin (EXLVI) is less than or equal to the detection voltage ( $V_{EXLVI}$ )) is generated and LVIIIF may be set to 1.

**(2) Low-voltage detection level select register (LVIS)**

This register selects the low-voltage detection level.

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

Reset signal generation input sets this register to 0EH.

**Figure 22-3. Format of Low-Voltage Detection Level Select Register (LVIS)**

Address: FFFAAH After reset: 0EH<sup>Note</sup> R/W

Symbol	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	Detection level
0	0	0	0	V <sub>LV10</sub> (4.22 ±0.1 V)
0	0	0	1	V <sub>LV11</sub> (4.07 ±0.1 V)
0	0	1	0	V <sub>LV12</sub> (3.92 ±0.1 V)
0	0	1	1	V <sub>LV13</sub> (3.76 ±0.1 V)
0	1	0	0	V <sub>LV14</sub> (3.61 ±0.1 V)
0	1	0	1	V <sub>LV15</sub> (3.45 ±0.1 V)
0	1	1	0	V <sub>LV16</sub> (3.30 ±0.1 V)
0	1	1	1	V <sub>LV17</sub> (3.15 ±0.1 V)
1	0	0	0	V <sub>LV18</sub> (2.99 ±0.1 V)
1	0	0	1	V <sub>LV19</sub> (2.84 ±0.1 V)
1	0	1	0	V <sub>LV110</sub> (2.68 ±0.1 V)
1	0	1	1	V <sub>LV111</sub> (2.53 ±0.1 V)
1	1	0	0	V <sub>LV112</sub> (2.38 ±0.1 V)
1	1	0	1	V <sub>LV113</sub> (2.22 ±0.1 V)
1	1	1	0	V <sub>LV114</sub> (2.07 ±0.1 V)
1	1	1	1	V <sub>LV115</sub> (1.91 ±0.1 V)

**Note** The reset value changes depending on the reset source.  
 If the LVIS register is reset by LVI, it is not reset but holds the current value. The value of this register is reset to “0EH” if a reset other than LVI is effected.

**Cautions 1. Be sure to clear bits 4 to 7 to “0”.**

**Cautions 2. Change the LVIS value with either of the following methods.**

- **When changing the value after stopping LVI**
    - <1> Stop LVI (LVION = 0).
    - <2> Change the LVIS register.
    - <3> Set to the mode used as an interrupt (LVIMD = 0).
    - <4> Mask LVI interrupts (LVIMK = 1).
    - <5> Enable LVI operation (LVION = 1).
    - <6> Before cancelling the LVI interrupt mask (LVIMK = 0), clear it with software because an LVIIF flag may be set when LVI operation is enabled.
  - **When changing the value after setting to the mode used as an interrupt (LVIMD = 0)**
    - <1> Mask LVI interrupts (LVIMK = 1).
    - <2> Set to the mode used as an interrupt (LVIMD = 0).
    - <3> Change the LVIS register.
    - <4> Before cancelling the LVI interrupt mask (LVIMK = 0), clear it with software because an LVIIF flag may be set when the LVIS register is changed.
- 3. When an input voltage from the external input pin (EXLVI) is detected, the detection voltage ( $V_{EXLVI}$ ) is fixed. Therefore, setting of LVIS is not necessary.**

**(3) Port mode register 12 (PM12)**

When using the P120/EXLVI/INTP0 pin for external low-voltage detection potential input, set PM120 to 1. At this time, the output latch of P120 may be 0 or 1.

PM12 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

**Figure 22-4. Format of Port Mode Register 12 (PM12)**

Address: FFF2CH    After reset: FFH    R/W

Symbol	7	6	5	4	3	2	1	0
PM12	1	1	1	1	1	1	1	PM120

PM120	P120 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

## 22.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

**(1) Used as reset (LVIMD = 1)**

- If LVISEL = 0, compares the supply voltage ( $V_{DD}$ ) and detection voltage ( $V_{LVI}$ ), generates an internal reset signal when  $V_{DD} < V_{LVI}$ , and releases internal reset when  $V_{DD} \geq V_{LVI}$ .
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage ( $V_{EXLVI}$ ), generates an internal reset signal when  $EXLVI < V_{EXLVI}$ , and releases internal reset when  $EXLVI \geq V_{EXLVI}$ .

**Remark** The low-voltage detector (LVI) can be set to ON by an option byte by default. If it is set to ON to raise the power supply from the POC detection voltage or lower, the internal reset signal is generated when the supply voltage ( $V_{DD}$ ) < detection voltage ( $V_{LVI} = 2.07 \text{ V} \pm 0.2 \text{ V}$ ). After that, the internal reset signal is generated when the supply voltage ( $V_{DD}$ ) < detection voltage ( $V_{LVI} = 2.07 \text{ V} \pm 0.1 \text{ V}$ ).

**(2) Used as interrupt (LVIMD = 0)**

- If LVISEL = 0, compares the supply voltage ( $V_{DD}$ ) and detection voltage ( $V_{LVI}$ ). When  $V_{DD}$  drops lower than  $V_{LVI}$  ( $V_{DD} < V_{LVI}$ ) or when  $V_{DD}$  becomes  $V_{LVI}$  or higher ( $V_{DD} \geq V_{LVI}$ ), generates an interrupt signal (INTLVI).
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage ( $V_{EXLVI} = 1.21 \text{ V} \pm 0.1 \text{ V}$ ). When EXLVI drops lower than  $V_{EXLVI}$  ( $EXLVI < V_{EXLVI}$ ) or when EXLVI becomes  $V_{EXLVI}$  or higher ( $EXLVI \geq V_{EXLVI}$ ), generates an interrupt signal (INTLVI).

While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

**Remark** LVIMD: Bit 1 of low-voltage detection register (LVIM)  
 LVISEL: Bit 2 of LVIM

### 22.4.1 When used as reset

#### (1) When detecting level of supply voltage (VDD)

(a) When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1)

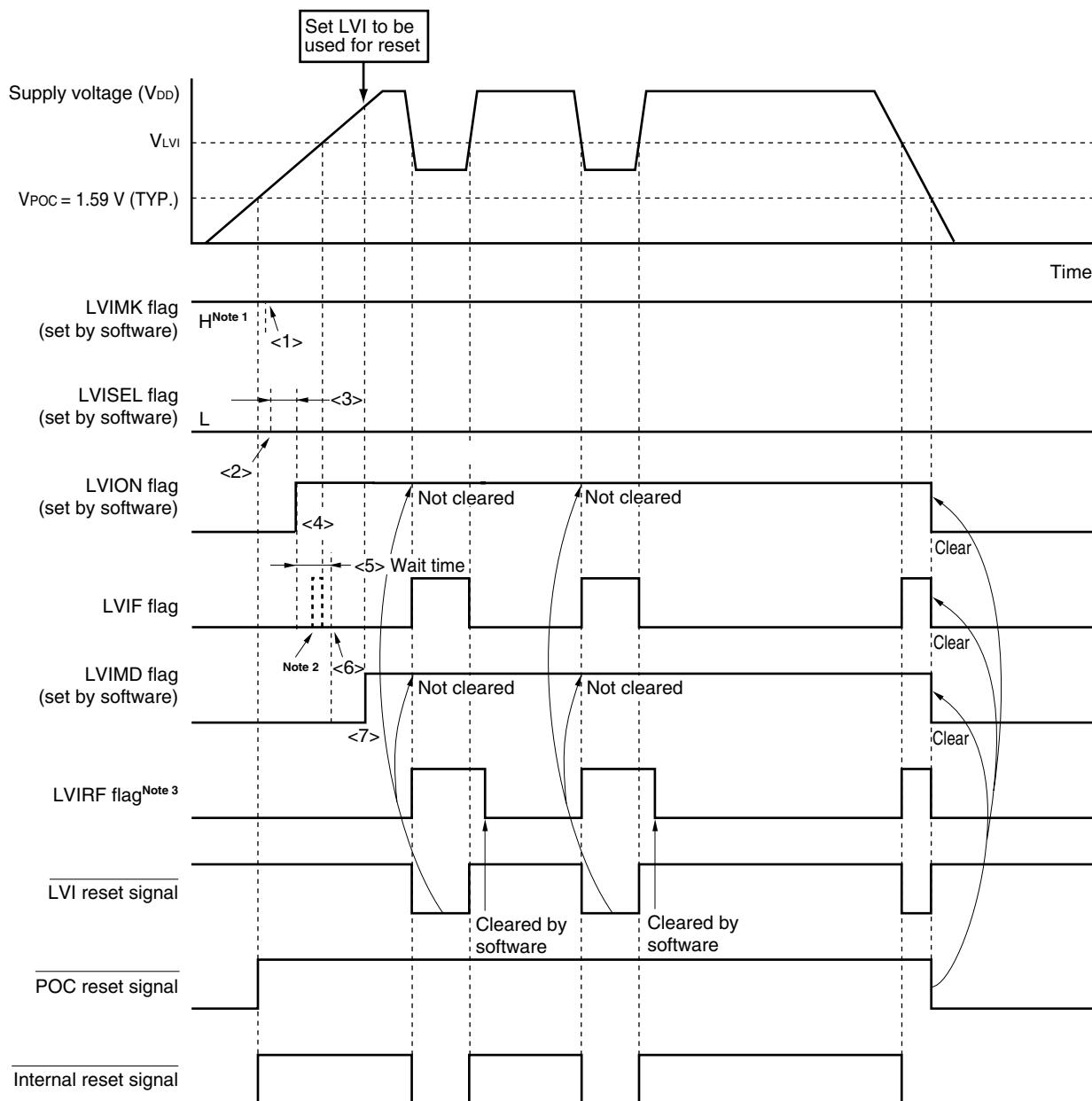
- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage ( $V_{DD}$ )) (default value).
  - <3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).
  - <4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <5> Use software to wait for the following periods of time (Total 410  $\mu$ s).
    - Operation stabilization time (10  $\mu$ s (MAX.))
    - Minimum pulse width (200  $\mu$ s (MIN.))
    - Detection delay time (200  $\mu$ s (MAX.))
  - <6> Wait until it is checked that (supply voltage ( $V_{DD}$ )  $\geq$  detection voltage ( $V_{LVI}$ )) by bit 0 (LVIF) of LVIM.
  - <7> Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected).

Figure 22-5 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

- Cautions**
1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <4>.
  2. If supply voltage ( $V_{DD}$ )  $\geq$  detection voltage ( $V_{LVI}$ ) when LVIMD is set to 1, an internal reset signal is not generated.

- When stopping operation
  - Either of the following procedures must be executed.
  - When using 8-bit memory manipulation instruction:  
Write 00H to LVIM.
  - When using 1-bit memory manipulation instruction:  
Clear LVIMD to 0 and then LVION to 0.

**Figure 22-5. Timing of Low-Voltage Detector Internal Reset Signal Generation**  
**(Bit: LVISEL = 0, Option Byte: LVIOFF = 1)**



- Notes**
1. The LVIMK flag is set to “1” by reset signal generation.
  2. The LVIF flag of the interrupt request flag registers and the LVIF flag may be set (1).
  3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 20 RESET FUNCTION**.

**Remark** <1> to <7> in Figure 22-5 above correspond to <1> to <7> in the description of “When starting operation” in 22.4.1 (1) (a) **When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1)**.



(b) When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0)

- When starting operation

Start in the following initial setting state.

- Set bit 7 (LVION) of LVIM to 1 (enables LVI operation)
- Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage ( $V_{DD}$ ))
- Set the low-voltage detection level selection register (LVIS) to 0EH (default value:  $V_{LVI} = 2.07 \text{ V} \pm 0.1 \text{ V}$ ).
- Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected)
- Set bit 0 (LVIF) of LVIM to 0 ("Supply voltage ( $V_{DD}$ )  $\geq$  detection voltage ( $V_{LVI}$ )")

Figure 22-6 shows the timing of the internal reset signal generated by the low-voltage detector.

- When stopping operation

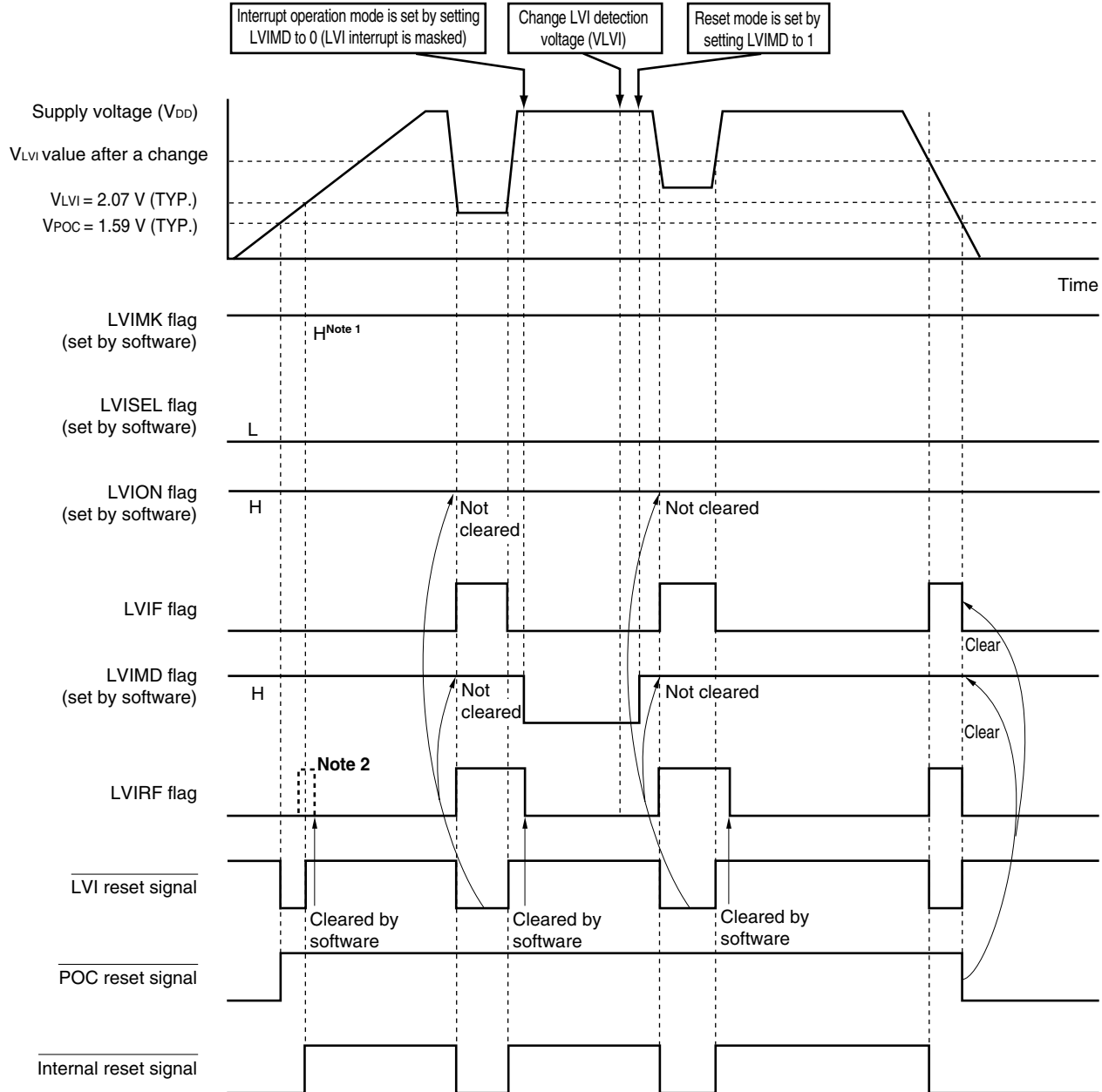
Either of the following procedures must be executed.

- When using 8-bit memory manipulation instruction:  
Write 00H to LVIM.
- When using 1-bit memory manipulation instruction:  
Clear LVIMD to 0 and then LVION to 0.

**Caution** Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- Does not perform low-voltage detection during LVION = 0.
- If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution. This is due to the fact that while the pulse width detected by LVI must be 200  $\mu\text{s}$  max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.

**Figure 22-6. Timing of Low-Voltage Detector Internal Reset Signal Generation**  
**(Bit: LVISEL = 0, Option Byte: LVIOFF = 0)**



**Notes 1.** The LVIMK flag is set to "1" by reset signal generation.

**2.** LVIRF is bit 0 of the reset control flag register (RESF).

When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, the LVIRF flag may become 1 from the beginning due to the power-on waveform.

For details of RESF, see **CHAPTER 20 RESET FUNCTION**.

**(2) When detecting level of input voltage from external input pin (EXLVI)**

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
  - <3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <4> Use software to wait for the following periods of time (Total 410  $\mu$ s).
    - Operation stabilization time (10  $\mu$ s (MAX.))
    - Minimum pulse width (200  $\mu$ s (MIN.))
    - Detection delay time (200  $\mu$ s (MAX.))
  - <5> Wait until it is checked that (input voltage from external input pin (EXLVI)  $\geq$  detection voltage ( $V_{EXLVI} = 1.21$  V (TYP.))) by bit 0 (LVIF) of LVIM.
  - <6> Set bit 1 (LVIMD) of LVIM to 1 (generates reset signal when the level is detected).

Figure 22-7 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <6> above.

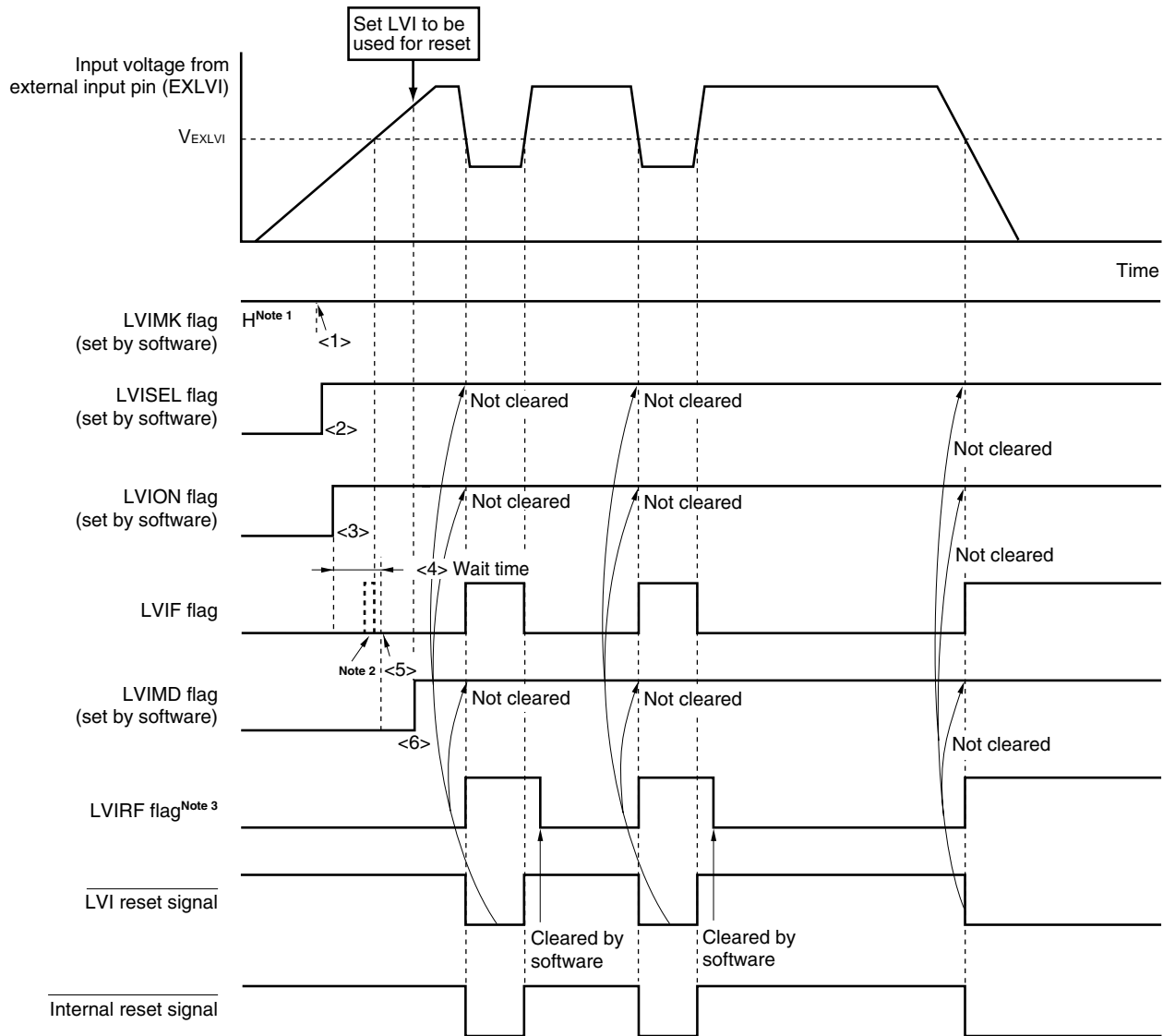
- Cautions**
1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.
  2. If input voltage from external input pin (EXLVI)  $\geq$  detection voltage ( $V_{EXLVI} = 1.21$  V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.
  3. Input voltage from external input pin (EXLVI) must be  $EXLVI < V_{DD}$ .

- When stopping operation
 

Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:  
Write 00H to LVIM.
  - When using 1-bit memory manipulation instruction:  
Clear LVIMD to 0 and then LVION to 0.

Figure 22-7. Timing of Low-Voltage Detector Internal Reset Signal Generation  
(Bit: LVISEL = 1)



- Notes**
1. The LVIMK flag is set to "1" by reset signal generation.
  2. The LVIF flag of the interrupt request flag registers and the LVIF flag may be set (1).
  3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 20 RESET FUNCTION**.

**Remark** <1> to <6> in Figure 22-7 above correspond to <1> to <6> in the description of "When starting operation" in 22.4.1 (2) **When detecting level of input voltage from external input pin (EXLVI)**.

## 22.4.2 When used as interrupt

### (1) When detecting level of supply voltage ( $V_{DD}$ )

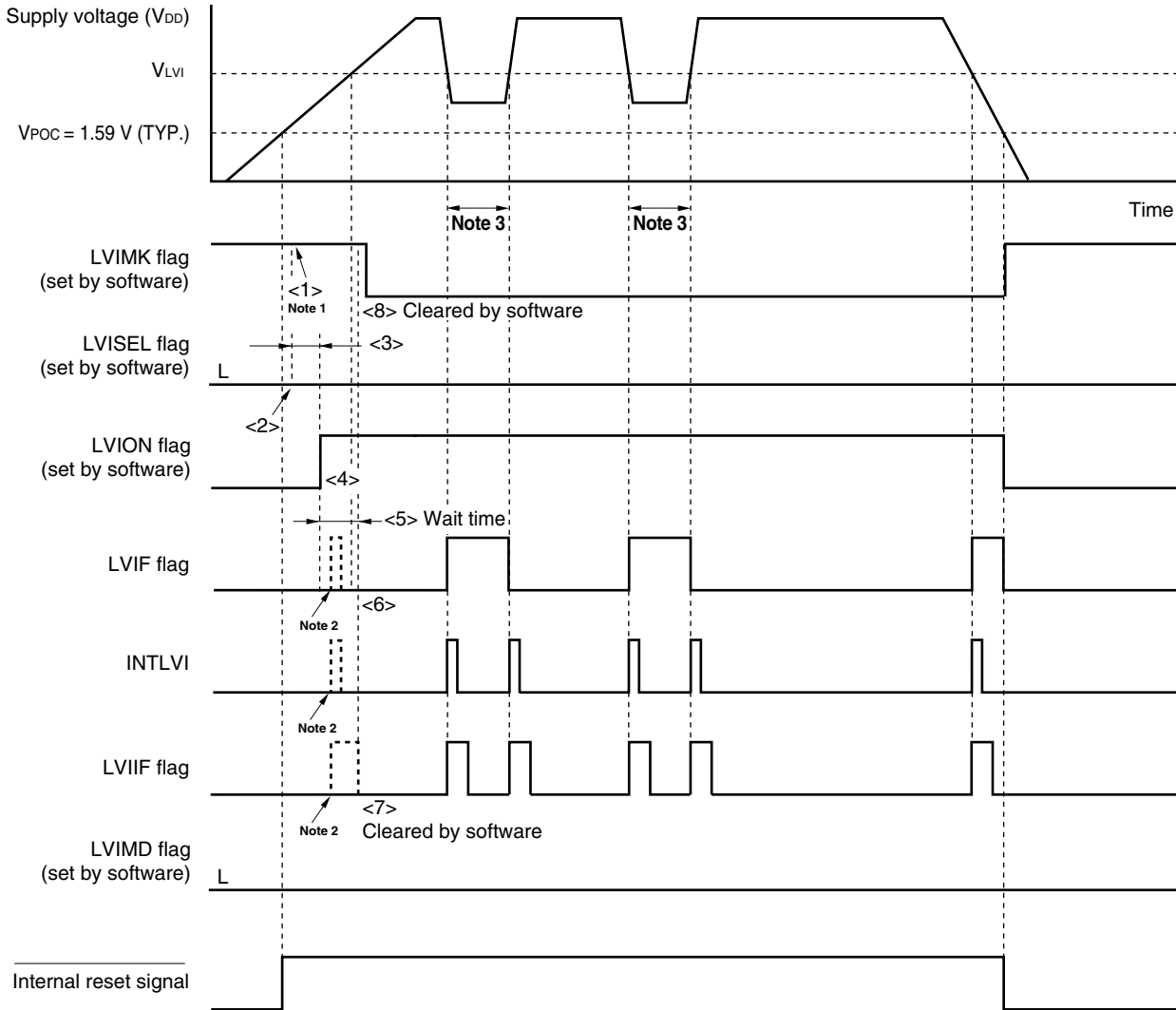
(a) When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1)

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage ( $V_{DD}$ )) (default value).  
Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).
  - <4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <5> Use software to wait for the following periods of time (Total 410  $\mu$ s).
    - Operation stabilization time (10  $\mu$ s (MAX.))
    - Minimum pulse width (200  $\mu$ s (MIN.))
    - Detection delay time (200  $\mu$ s (MAX.))
  - <6> Confirm that “supply voltage ( $V_{DD}$ )  $\geq$  detection voltage ( $V_{LVI}$ )” when detecting the falling edge of  $V_{DD}$ , or “supply voltage ( $V_{DD}$ ) < detection voltage ( $V_{LVI}$ )” when detecting the rising edge of  $V_{DD}$ , at bit 0 (LVIF) of LVIM.
  - <7> Clear the interrupt request flag of LVI (LVIIF) to 0.
  - <8> Release the interrupt mask flag of LVI (LVIMK).
  - <9> Execute the EI instruction (when vector interrupts are used).

Figure 22-8 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <8> above.

- When stopping operation
  - Either of the following procedures must be executed.
  - When using 8-bit memory manipulation instruction:  
Write 00H to LVIM.
  - When using 1-bit memory manipulation instruction:  
Clear LVION to 0.

**Figure 22-8. Timing of Low-Voltage Detector Interrupt Signal Generation**  
**(Bit: LVISEL = 0, Option Byte: LVIOFF = 1)**



- Notes**
1. The LVIMK flag is set to “1” by reset signal generation.
  2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).
  3. If LVI operation is disabled when the supply voltage (V<sub>DD</sub>) is less than or equal to the detection voltage (V<sub>LVI</sub>), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

**Remark** <1> to <8> in Figure 22-8 above correspond to <1> to <8> in the description of “When starting operation” in 22.4.2 (1) (a) **When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1)**.

(b) When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0)

- When starting operation
  - <1> Start in the following initial setting state.
    - Set bit 7 (LVION) of LVIM to 1 (enables LVI operation)
    - Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage ( $V_{DD}$ ))
    - Set the low-voltage detection level selection register (LVIS) to 0EH (default value:  $V_{LVI} = 2.07\text{ V} \pm 0.1\text{ V}$ ).
    - Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected)
    - Set bit 0 (LVIF) of LVIM to 0 (Detects falling edge “Supply voltage ( $V_{DD}$ )  $\geq$  detection voltage ( $V_{LVI}$ )”)
  - <2> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <3> Release the interrupt mask flag of LVI (LVIMK).
  - <4> Execute the EI instruction (when vector interrupts are used).

Figure 22-9 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <3> above.

- When stopping operation
 

Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
 

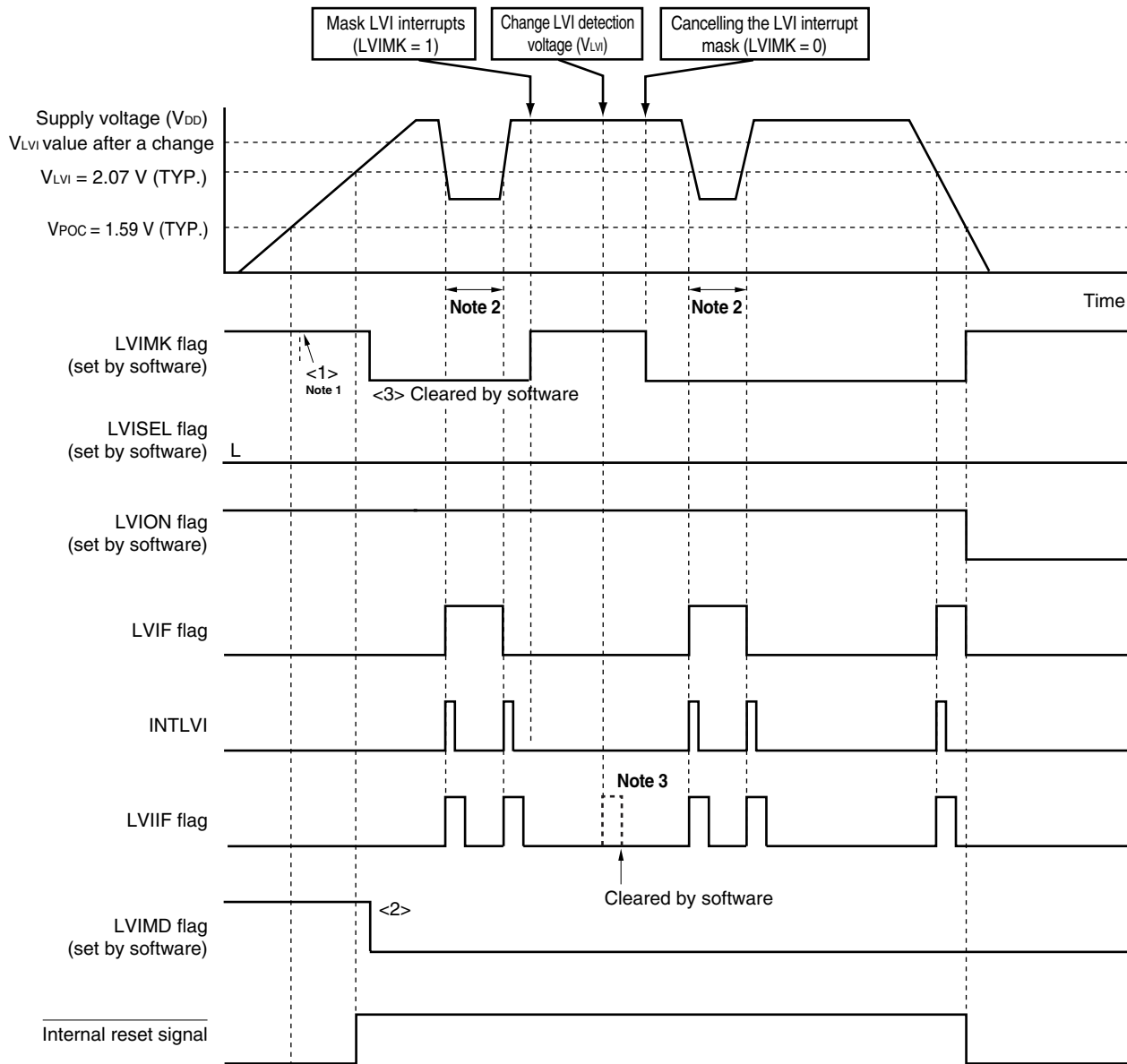
Write 00H to LVIM.
  - When using 1-bit memory manipulation instruction:
 

Clear LVION to 0.

**Cautions 1. Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:**

- Does not perform low-voltage detection during LVION = 0.
  - If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution. This is due to the fact that while the pulse width detected by LVI must be 200  $\mu\text{s}$  max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.
- 2. When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, the LVIRF flag may become 1 from the beginning due to the power-on waveform.**  
For details of RESF, see CHAPTER 20 RESET FUNCTION.

**Figure 22-9. Timing of Low-Voltage Detector Interrupt Signal Generation**  
**(Bit: LVISEL = 0, Option Byte: LVIOFF = 0)**



- Notes**
1. The LVIMK flag is set to “1” by reset signal generation.
  2. If LVI operation is disabled when the supply voltage ( $V_{DD}$ ) is less than or equal to the detection voltage ( $V_{LVI}$ ), an interrupt request signal (INTLVI) is generated and LVIF may be set to 1.
  3. The LVIIF flag may be set when the LVI detection voltage is changed.

**Remark** <1> to <3> in Figure 22-9 above correspond to <1> to <3> in the description of “When starting operation” in 22.4.2 (1) (b) When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0).



**(2) When detecting level of input voltage from external input pin (EXLVI)**

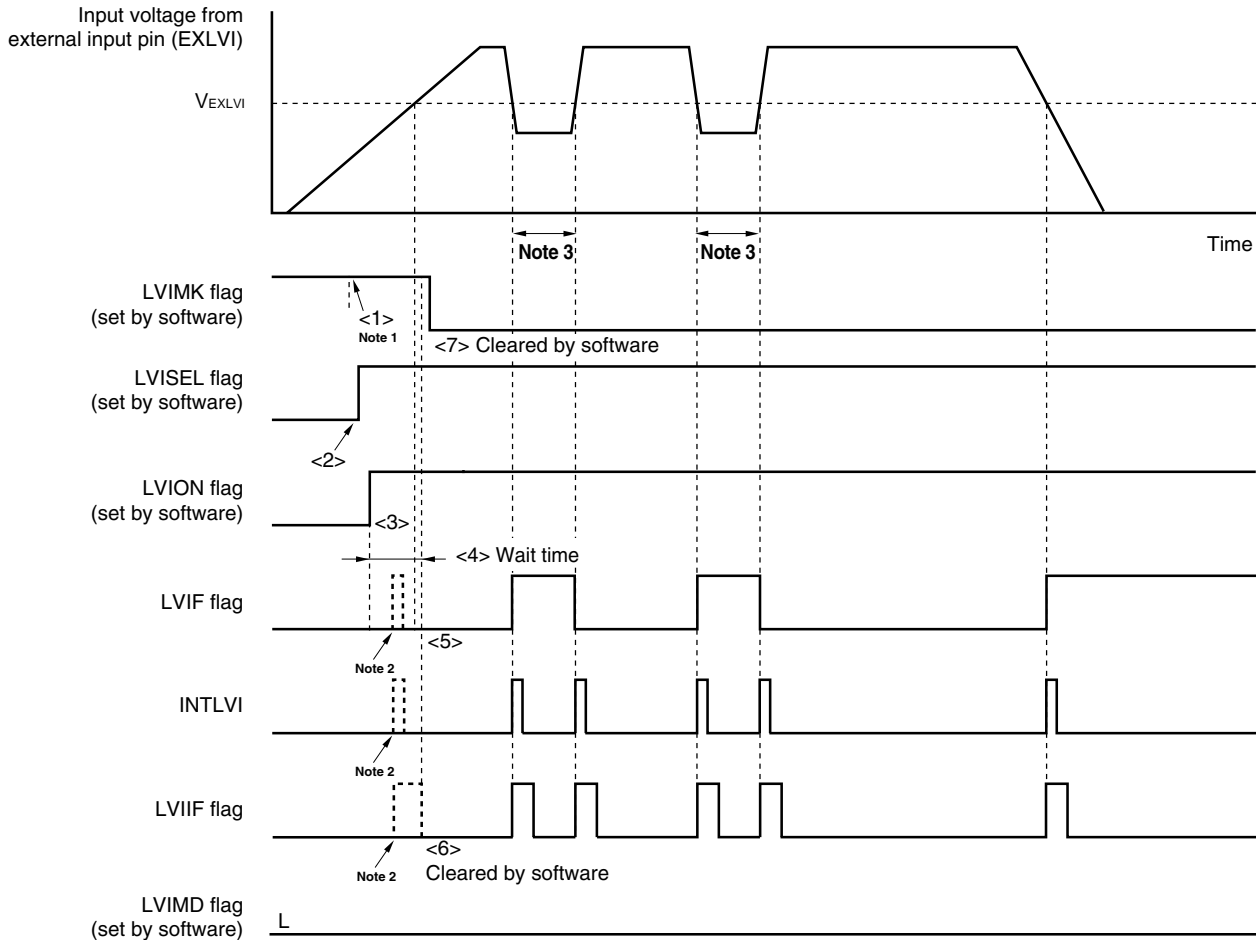
- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).  
Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <4> Use software to wait for the following periods of time (Total 410  $\mu$ s).
    - Operation stabilization time (10  $\mu$ s (MAX.))
    - Minimum pulse width (200  $\mu$ s (MIN.))
    - Detection delay time (200  $\mu$ s (MAX.))
  - <5> Confirm that “input voltage from external input pin (EXLVI)  $\geq$  detection voltage ( $V_{EXLVI} = 1.21$  V (TYP.))” when detecting the falling edge of EXLVI, or “input voltage from external input pin (EXLVI)  $<$  detection voltage ( $V_{EXLVI} = 1.21$  V (TYP.))” when detecting the rising edge of EXLVI, at bit 0 (LVIF) of LVIM.
  - <6> Clear the interrupt request flag of LVI (LVIF) to 0.
  - <7> Release the interrupt mask flag of LVI (LVIMK).
  - <8> Execute the EI instruction (when vector interrupts are used).

Figure 22-10 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

**Caution** Input voltage from external input pin (EXLVI) must be  $EXLVI < V_{DD}$ .

- When stopping operation  
Either of the following procedures must be executed.
  - When using 8-bit memory manipulation instruction:  
Write 00H to LVIM.
  - When using 1-bit memory manipulation instruction:  
Clear LVION to 0.

**Figure 22-10. Timing of Low-Voltage Detector Interrupt Signal Generation  
(Bit: LVISEL = 1)**



- Notes**
1. The LVIMK flag is set to “1” by reset signal generation.
  2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).
  3. If LVI operation is disabled when the input voltage of external input pin (EXLVI) is less than or equal to the detection voltage ( $V_{EXLVI}$ ), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

**Remark** <1> to <7> in Figure 22-10 above correspond to <1> to <7> in the description of “When starting operation” in 22.4.2 (2) **When detecting level of input voltage from external input pin (EXLVI).**

## 22.5 Cautions for Low-Voltage Detector

### (1) Measures method when supply voltage ( $V_{DD}$ ) frequently fluctuates in the vicinity of the LVI detection voltage ( $V_{LVI}$ )

In a system where the supply voltage ( $V_{DD}$ ) fluctuates for a certain period in the vicinity of the LVI detection voltage ( $V_{LVI}$ ), the operation is as follows depending on how the low-voltage detector is used.

#### Operation example 1: When used as reset

The system may be repeatedly reset and released from the reset status.

The time from reset release through microcontroller operation start can be set arbitrarily by the following action.

#### <Action>

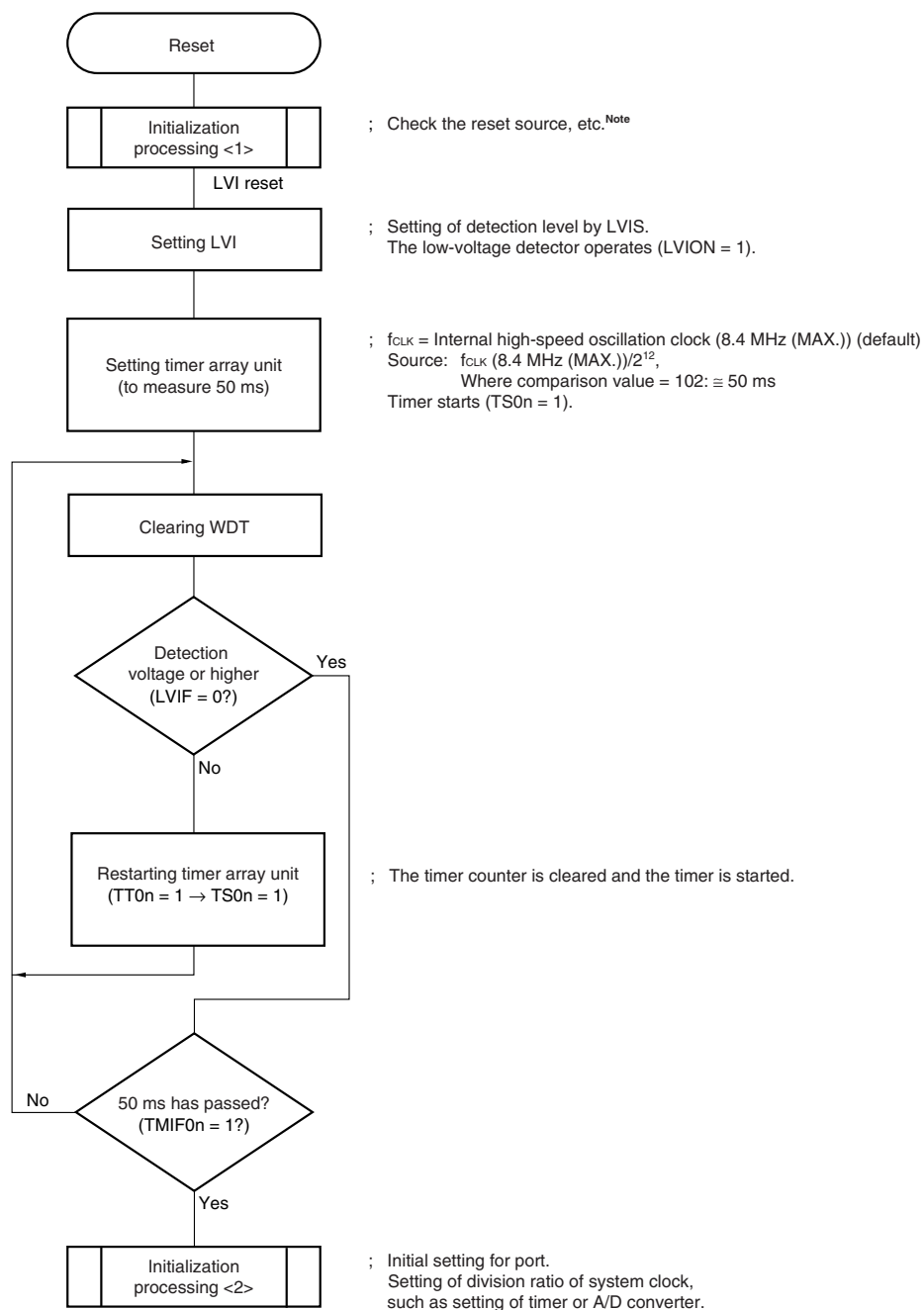
After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (see **Figure 22-11**).

**Remark** If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to “1”, the meanings of the above words change as follows.

- Supply voltage ( $V_{DD}$ ) → Input voltage from external input pin (EXLVI)
- Detection voltage ( $V_{LVI}$ ) → Detection voltage ( $V_{EXLVI} = 1.21$  V)

Figure 22-11. Example of Software Processing After Reset Release (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



**Note** A flowchart is shown on the next page.

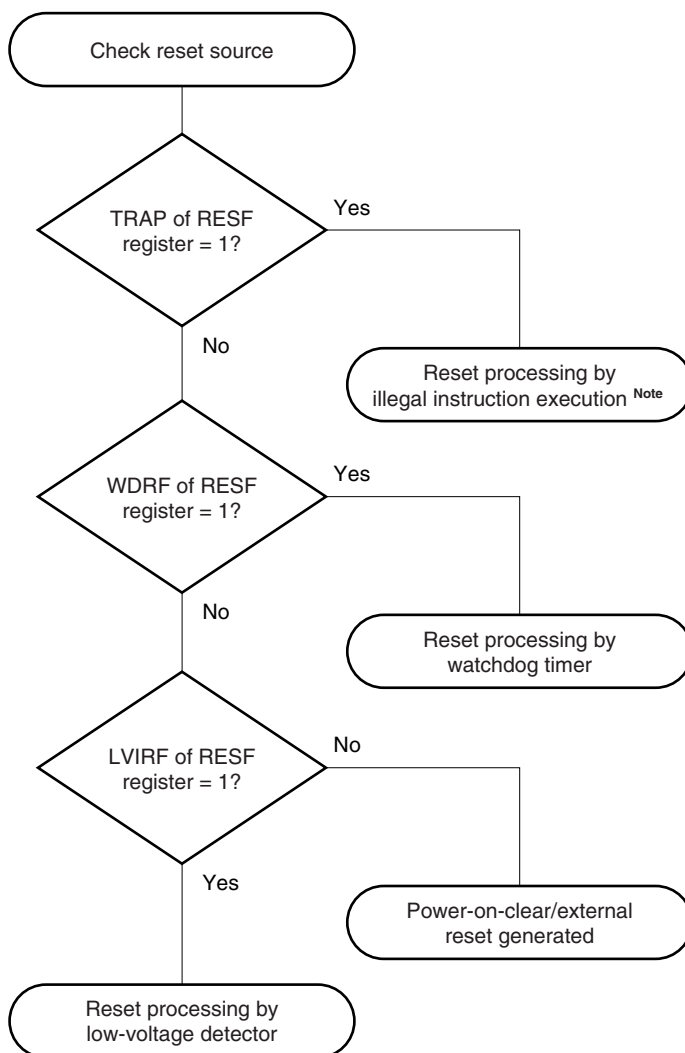
**Remarks 1.** n: Channel number (n = 0 to 7)

**2.** If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to “1”, the meanings of the above words change as follows.

- Supply voltage (V<sub>DD</sub>) → Input voltage from external input pin (EXLVI)
- Detection voltage (V<sub>LVI</sub>) → Detection voltage (V<sub>EXLVI</sub> = 1.21 V)

Figure 22-11. Example of Software Processing After Reset Release (2/2)

- Checking reset source



**Note** When instruction code FFH is executed.

Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

**Remark** If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to “1”, the meanings of the above words change as follows.

- Supply voltage ( $V_{DD}$ ) → Input voltage from external input pin (EXLVI)
- Detection voltage ( $V_{LVI}$ ) → Detection voltage ( $V_{EXLVI} = 1.21\text{ V}$ )

**Operation example 2: When used as interrupt**

Interrupt requests may be generated frequently.

Take the following action.

**<Action>**

Confirm that “supply voltage ( $V_{DD}$ )  $\geq$  detection voltage ( $V_{LVI}$ )” when detecting the falling edge of  $V_{DD}$ , or “supply voltage ( $V_{DD}$ )  $<$  detection voltage ( $V_{LVI}$ )” when detecting the rising edge of  $V_{DD}$ , in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 1 (LVIIF) of interrupt request flag register 0L (IF0L) to 0.

For a system with a long supply voltage fluctuation period near the LVI detection voltage, take the above action after waiting for the supply voltage fluctuation time.

**Remark** If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to “1”, the meanings of the above words change as follows.

- Supply voltage ( $V_{DD}$ ) → Input voltage from external input pin (EXLVI)
- Detection voltage ( $V_{LVI}$ ) → Detection voltage ( $V_{EXLVI} = 1.21$  V)

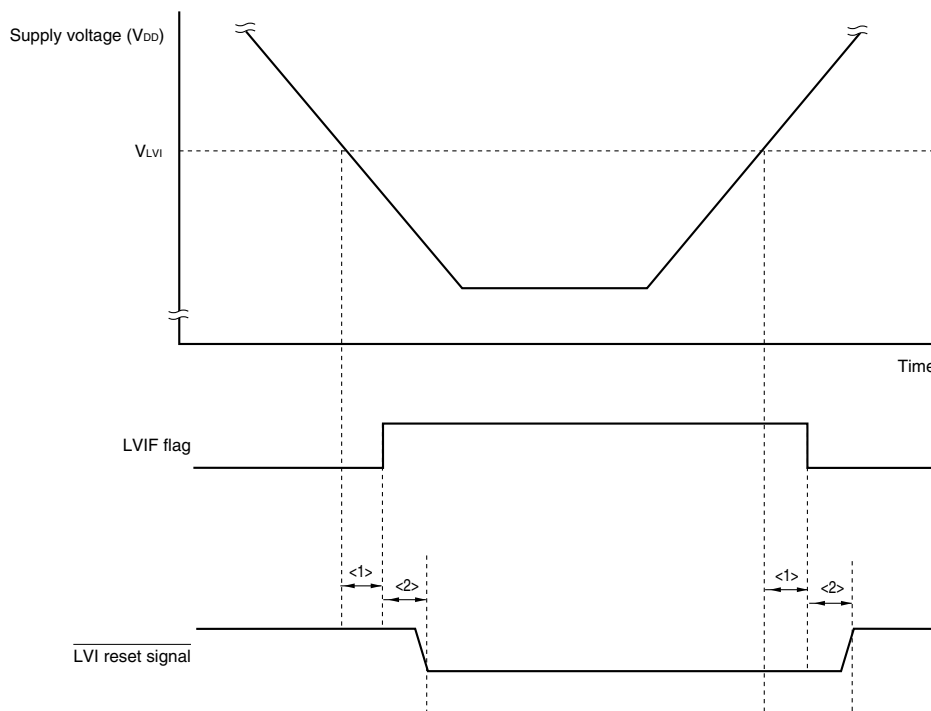
**(2) Delay from the time LVI reset source is generated until the time LVI reset has been generated or released**

There is some delay from the time supply voltage ( $V_{DD}$ )  $<$  LVI detection voltage ( $V_{LVI}$ ) until the time LVI reset has been generated.

In the same way, there is also some delay from the time LVI detection voltage ( $V_{LVI}$ )  $\leq$  supply voltage ( $V_{DD}$ ) until the time LVI reset has been released (see **Figure 22-12**).

See the timing in **Figure 21-2 (2) When LVI is ON upon power application (option byte: LVIOFF = 0)** for the reset processing time until the normal operation is entered after the LVI reset is released.

**Figure 22-12. Delay from the time LVI reset source is generated until the time LVI reset has been generated or released**



<1> : Minimum pulse width (200  $\mu$ s (MIN.))

<2> : Detection delay time (200  $\mu$ s (MAX.))

## CHAPTER 23 REGULATOR

### 23.1 Regulator Overview

The 78K0R/KH3 contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to  $V_{SS}$  via a capacitor (0.47 to 1  $\mu\text{F}$ ). However, when using the STOP mode that has been entered since operation of the internal high-speed oscillation clock and external main system clock, 0.47  $\mu\text{F}$  is recommended. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

The regulator output voltage is normally 2.5 V (typ.), and in the low consumption current mode, 1.8 V (typ.).

### 23.2 Registers Controlling Regulator

#### (1) Regulator mode control register (RMC)

This register sets the output voltage of the regulator.

RMC is set with an 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

**Figure 23-1. Format of Regulator Mode Control Register (RMC)**

Address: F00F4H    After reset: 00H    R/W

Symbol	7	6	5	4	3	2	1	0
RMC								

RMC[7:0]	Control of output voltage of regulator
5AH	Fixed to low consumption current mode (1.8 V)
00H	Switches normal current mode (2.5 V) and low consumption current mode (1.8 V) according to the condition (refer to <b>Table 23-1</b> )
Other than above	Setting prohibited

- Cautions**
1. The RMC register can be rewritten only in the low consumption current mode (refer to Table 23-1). In other words, rewrite this register during CPU operation with the subsystem clock ( $f_{XT}$ ) while the high-speed system clock ( $f_{MX}$ ) and high-speed internal oscillation clock ( $f_{IH}$ ) are both stopped.
  2. When using the setting fixed to the low consumption current mode, the RMC register can be used in the following cases.
    - <When X1 clock is selected as the CPU clock>                       $f_X \leq 5 \text{ MHz}$  and  $f_{CLK} \leq 5 \text{ MHz}$
    - <When the high-speed internal oscillation clock, external input clock, or subsystem clock are selected for the CPU clock>                       $f_{CLK} \leq 5 \text{ MHz}$
  3. The self-programming function is disabled in the low consumption current mode.

Table 23-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
Low consumption current mode	1.8 V	During system reset
		In STOP mode (except during OCD mode)
		When both the high-speed system clock ( $f_{MX}$ ) and the high-speed internal oscillation clock ( $f_{IH}$ ) are stopped during CPU operation with the subsystem clock ( $f_{XT}$ )
		When both the high-speed system clock ( $f_{MX}$ ) and the high-speed internal oscillation clock ( $f_{IH}$ ) are stopped during the HALT mode when the CPU operation with the subsystem clock ( $f_{XT}$ ) has been set
Normal current mode	2.5 V	Other than above



## CHAPTER 24 OPTION BYTE

### 24.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the 78K0R/KH3 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.

To use the boot swap operation during self programming, 000C0H to 000C3H are replaced by 010C0H to 010C3H. Therefore, set the same values as 000C0H to 000C3H to 010C0H to 010C3H.

**Caution** Be sure to set FFH to 000C2H (000C2H/010C2H when the boot swap operation is used).

#### 24.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)

##### (1) 000C0H/010C0H

- Operation of watchdog timer
  - Operation is stopped or enabled in the HALT or STOP mode.
- Setting of interval time of watchdog timer
- Operation of watchdog timer
  - Operation is stopped or enabled.
- Setting of window open period of watchdog timer
- Setting of interval interrupt of watchdog timer
  - Used or not used

**Caution** Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

##### (2) 000C1H/010C1H

- Setting of LVI upon reset release (upon power application)
  - LVI is ON or OFF by default upon reset release (reset by  $\overline{\text{RESET}}$  pin excluding LVI, POC, WDT, or illegal instructions).

**Caution** Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

##### (3) 000C2H/010C2H

- Be sure to set FFH, as these addresses are reserved areas.

**Caution** Set FFH to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

24.1.2 On-chip debug option byte (000C3H/ 010C3H)

- Control of on-chip debug operation
  - On-chip debug operation is disabled or enabled.
- Handling of data of flash memory in case of failure in on-chip debug security ID authentication
  - Data of flash memory is erased or not erased in case of failure in on-chip debug security ID authentication.

**Caution** Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

24.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 24-1. Format of User Option Byte (000C0H/010C0H) (1/2)

Address: 000C0H/010C0H<sup>Note 1</sup>

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINIT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period <sup>Note 2</sup>
0	0	25%
0	1	50%
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time
0	0	0	$2^{10}/f_{IL}$ (3.88 ms)
0	0	1	$2^{11}/f_{IL}$ (7.76 ms)
0	1	0	$2^{12}/f_{IL}$ (15.52 ms)
0	1	1	$2^{13}/f_{IL}$ (31.03 ms)
1	0	0	$2^{15}/f_{IL}$ (124.12 ms)
1	0	1	$2^{17}/f_{IL}$ (496.48 ms)
1	1	0	$2^{18}/f_{IL}$ (992.97 ms)
1	1	1	$2^{20}/f_{IL}$ (3971.88 ms)

Figure 24-1. Format of User Option Byte (000C0H/010C0H) (2/2)

Address: 000C0H/010C0H<sup>Note 1</sup>

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON
WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)						
0	Counter operation stopped in HALT/STOP mode <sup>Note 2</sup>						
1	Counter operation enabled in HALT/STOP mode						

- Notes**
1. Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.
  2. The window open period is 100% when WDSTBYON = 0, regardless the value of WINDOW1 and WINDOW0.

**Caution** The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

- Remarks**
1.  $f_{IL}$ : Internal low-speed oscillation clock frequency
  2. ( ):  $f_{IL} = 264 \text{ kHz (MAX.)}$

Figure 24-2. Format of Option Byte (000C1H/010C1H)

Address: 000C1H/010C1H<sup>Note</sup>

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	LVIOFF
LVIOFF	Setting of LVI on power application						
0	LVI is ON by default (LVI default start function enabled) upon reset release (upon power application)						
1	LVI is OFF by default (LVI default start function stopped) upon reset release (upon power application)						

**Note** Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

- Cautions**
1. Be sure to set bits 7 to 1 to “1”.
  2. Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:
    - Does not perform low-voltage detection during LVION = 0.
    - If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution. This is due to the fact that while the pulse width detected by LVI must be 200  $\mu\text{s}$  max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.

**Figure 24-3. Format of Option Byte (000C2H/010C2H)**Address: 000C2H/010C2H<sup>Note</sup>

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

**Note** Be sure to set FFH to 000C2H, as these addresses are reserved areas. Also set FFH to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

### 24.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

**Figure 24-4. Format of On-chip Debug Option Byte (000C3H/010C3H)**Address: 000C3H/010C3H<sup>Note</sup>

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Enables on-chip debugging. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debugging. Does not erases data of flash memory in case of failures in authenticating on-chip debug security ID.

**Note** Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

**Caution** Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value.  
Be sure to set 000010B to bits 6 to 1.

**Remark** The value on bits 3 to 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.  
However, be sure to set the default values (0, 1, and 0) to bits 3 to 1 at setting.

## 24.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the RA78K0R or PM+ linker option, in addition to describing to the source. When doing so, the contents set by using the linker option take precedence, even if descriptions exist in the source, as mentioned below.

See the **RA78K0R Assembler Package User's Manual** for how to set the linker option.

A software description example of the option byte setting is shown below.

OPT	CSEG	OPT_BYTE	
	DB	10H	; Does not use interval interrupt of watchdog timer, ; Enables watchdog timer operation, ; Window open period of watchdog timer is 25%, ; Overflow time of watchdog timer is $2^{10}/f_{IL}$ , ; Stops watchdog timer operation during HALT/STOP mode
	DB	0FFH	; Stops LVI default start function
	DB	0FFH	; Reserved area
	DB	85H	; Enables on-chip debug operation, does not erase flash memory ; data when security ID authorization fails

When the boot swap function is used during self programming, 000C0H to 000C3H is switched to 010C0H to 010C3H. Describe to 010C0H to 010C3H, therefore, the same values as 000C0H to 000C3H as follows.

OPT2	CSEG	AT	010C0H	
	DB		10H	; Does not use interval interrupt of watchdog timer, ; Enables watchdog timer operation, ; Window open period of watchdog timer is 25%, ; Overflow time of watchdog timer is $2^{10}/f_{IL}$ , ; Stops watchdog timer operation during HALT/STOP mode
	DB		0FFH	; Stops LVI default start function
	DB		0FFH	; Reserved area
	DB		85H	; Enables on-chip debug operation, does not erase flash memory ; data when security ID authorization fails

**Caution** To specify the option byte by using assembly language, use **OPT\_BYTE** as the relocation attribute name of the CSEG pseudo instruction. To specify the option byte to 010C0H to 010C3H in order to use the boot swap function, use the relocation attribute **AT** to specify an absolute address.

## CHAPTER 25 FLASH MEMORY

The 78K0R/KH3 incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board.

### 25.1 Writing with Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the 78K0R/KH3.

- PG-FP4, FL-PR4
- PG-FP5, FL-PR5
- QB-MINI2

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 78K0R/KH3 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 78K0R/KH3 is mounted on the target system.

**Remark** The FL-PR4, FL-PR5, and FA series are products of Naito Densai Machida Mfg. Co., Ltd.

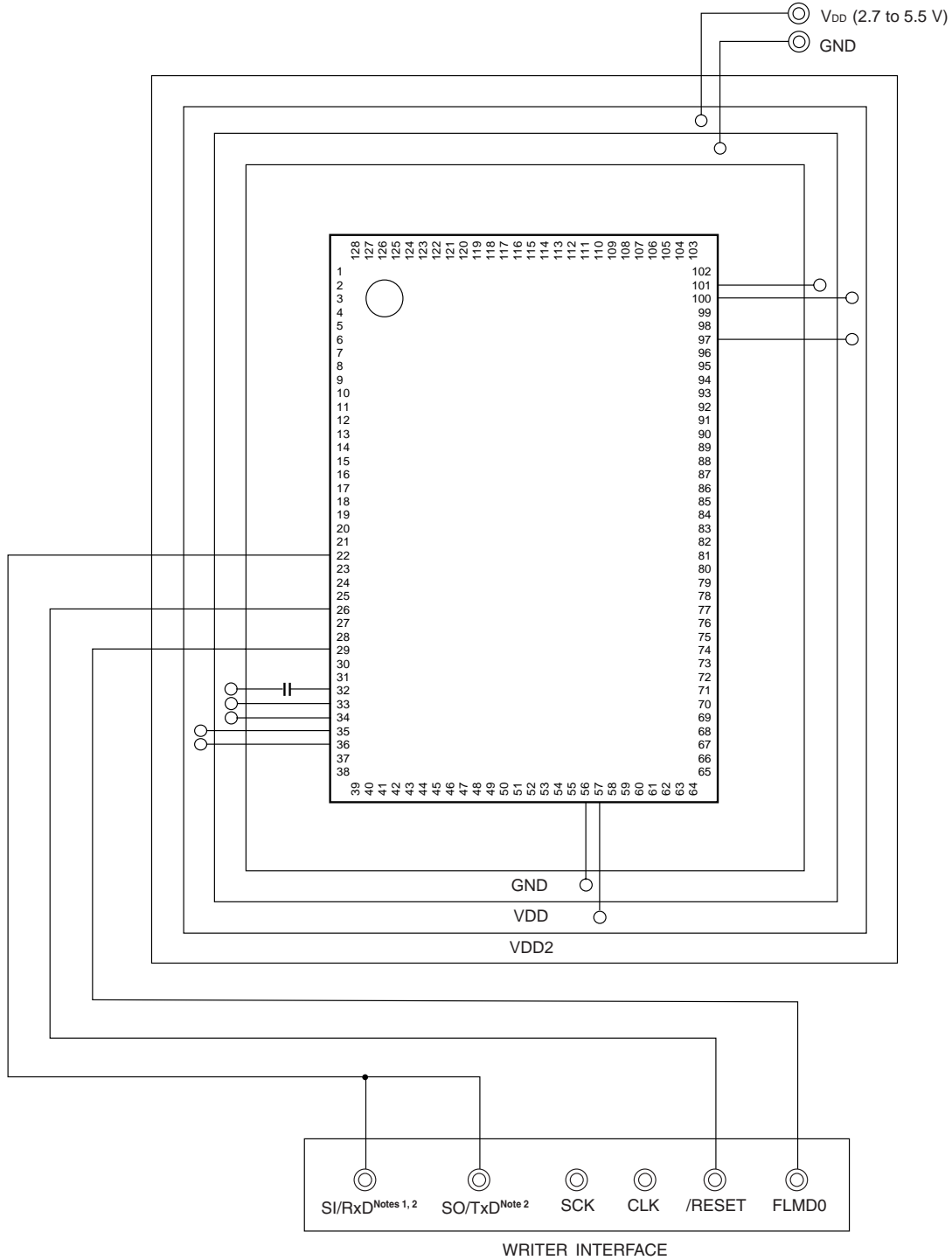
**Table 25-1. Wiring Between 78K0R/KH3 and Dedicated Flash Memory Programmer**

Pin Configuration of Dedicated Flash Memory Programmer			Pin Name	Pin No.
Signal Name	I/O	Pin Function		
SI/RxD <sup>Notes 1, 2</sup>	Input	Receive signal	TOOLO/P40	22
SO/TxD <sup>Note 2</sup>	Output	Transmit signal		
SCK	Output	Transfer clock	–	–
CLK	Output	Clock output	–	–
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	26
FLMD0	Output	Mode signal	FLMD0	29
V <sub>DD</sub>	I/O	V <sub>DD</sub> voltage generation/ power monitoring	V <sub>DD</sub>	35
			EV <sub>DD0</sub>	36
			EV <sub>DD1</sub>	57
			AV <sub>REF0</sub>	100
			AV <sub>REF1</sub>	97
GND	–	Ground	V <sub>SS</sub>	33
			EV <sub>SS0</sub>	34
			EV <sub>SS1</sub>	56
			AV <sub>SS</sub>	101

- Notes**
1. This pin is not required to be connected when using PG-FP5 or FL-PR5.
  2. Connect SI/RxD or SO/TxD when using QB-MINI2.

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

**Figure 25-1. Example of Wiring Adapter for Flash Memory Writing**



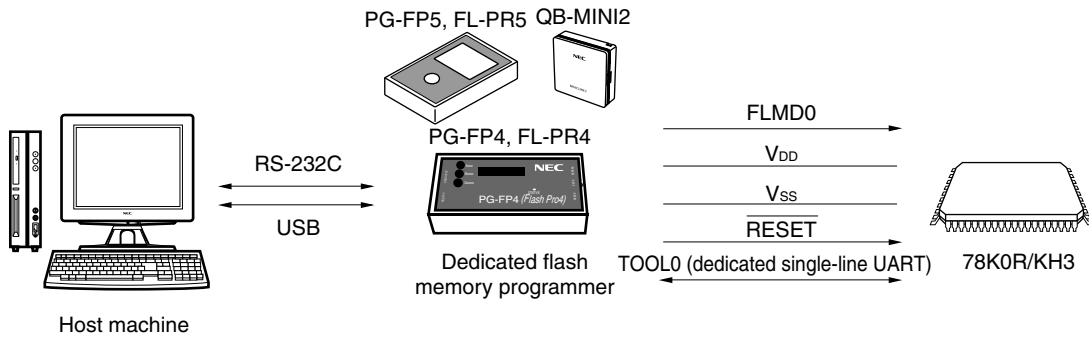
- Notes**
1. This pin is not required to be connected when using PG-FP5 or FL-PR5.
  2. Connect SI/RxD or SO/TxD when using QB-MINI2.



## 25.2 Programming Environment

The environment required for writing a program to the flash memory of the 78K0R/KH3 is illustrated below.

**Figure 25-2. Environment for Writing Program to Flash Memory**



A host machine that controls the dedicated flash memory programmer is necessary.

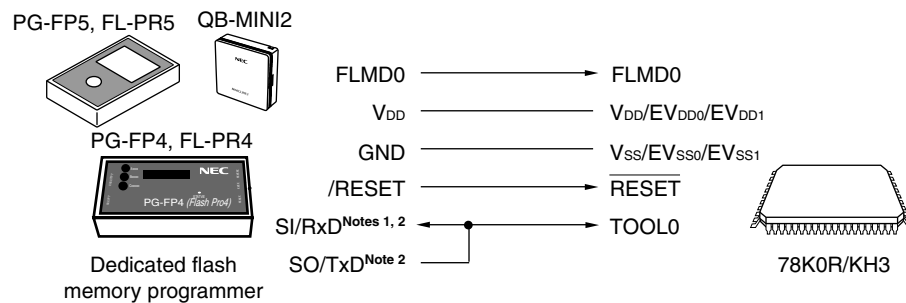
To interface between the dedicated flash memory programmer and the 78K0R/KH3, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

## 25.3 Communication Mode

Communication between the dedicated flash memory programmer and the 78K0R/KH3 is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the 78K0R/KH3.

Transfer rate: 115,200 bps to 1,000,000 bps

**Figure 25-3. Communication with Dedicated Flash Memory Programmer**



- Notes**
1. This pin is not required to be connected when using PG-FP5 or FL-PR5.
  2. Connect SI/RxD or SO/TxD when using QB-MINI2.

The dedicated flash memory programmer generates the following signals for the 78K0R/KH3. See the manual of PG-FP4, FL-PR4, PG-FP5, FL-PR5, or MINICUBE2 for details.

When using the FlashPro4 as the dedicated flash memory programmer, the FlashPro4 generates the following signals for the 78K0R/KH3. For details, refer to the user's manual for the FlashPro4.

**Table 25-2. Pin Connection**

Dedicated Flash Memory Programmer			78K0R/KH3	Connection
Signal Name	I/O	Pin Function	Pin Name	
FLMD0	Output	Mode signal	FLMD0	⊙
V <sub>DD</sub>	I/O	V <sub>DD</sub> voltage generation/power monitoring	V <sub>DD</sub> , EV <sub>DD0</sub> , EV <sub>DD1</sub> , AV <sub>REF0</sub> , AV <sub>REF1</sub>	⊙
GND	–	Ground	V <sub>SS</sub> , EV <sub>SS0</sub> , EV <sub>SS1</sub> , AV <sub>SS</sub>	⊙
CLK	Output	Clock output	–	×
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	⊙
SI/RxD <sup>Notes 1, 2</sup>	Input	Receive signal	TOOL0	⊙
SO/TxD <sup>Note 2</sup>	Output	Transmit signal		
SCK	Output	Transfer clock	–	×

**Notes** 1. This pin is not required to be connected when using PG-FP5 or FL-PR5.

2. Connect SI/RxD or SO/TxD when using QB-MINI2.

**Remark** ⊙: Be sure to connect the pin.

×: The pin does not have to be connected.

## 25.4 Connection of Pins on Board

To write the flash memory on-board, 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.

### 25.4.1 FLMD0 pin

#### (1) In flash memory programming mode

Directly connect this pin to a flash memory programmer when data is written by the flash memory programmer.

This supplies a writing voltage of the V<sub>DD</sub> level to the FLMD0 pin.

The FLMD0 pin does not have to be pulled down externally because it is internally pulled down by reset. To pull it down externally, use a resistor of 1 kΩ to 200 kΩ.

#### (2) In normal operation mode

It is recommended to leave this pin open during normal operation.

The FLMD0 pin must always be kept at the V<sub>SS</sub> level before reset release but does not have to be pulled down externally because it is internally pulled down by reset. However, pulling it down must be kept selected (i.e., FLMDPUP = "0", default value) by using bit 7 (FLMDPUP) of the background event control register (BECTL) (see **25.5 (1) Back ground event control register**). To pull it down externally, use a resistor of 200 kΩ or smaller.

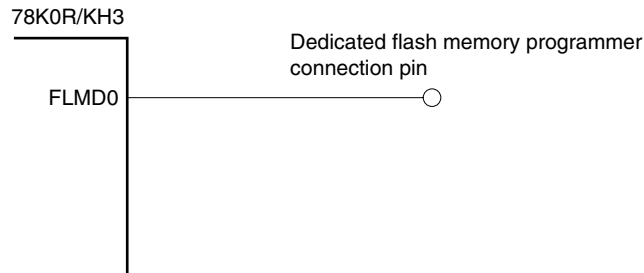
Self programming and the rewriting of flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V<sub>SS</sub> pin.

**(3) In self programming mode**

It is recommended to leave this pin open when using the self programming function. To pull it down externally, use a resistor of 100 k $\Omega$  to 200 k $\Omega$ .

In the self programming mode, the setting is switched to pull up in the self programming library.

**Figure 25-4. FLMD0 Pin Connection Example**

**25.4.2 TOOL0 pin**

In the flash memory programming mode, connect this pin directly to the dedicated flash memory programmer or pull it up by connecting it to EV<sub>DD0</sub> or EV<sub>DD1</sub> via an external resistor.

When on-chip debugging is enabled in the normal operation mode, pull this pin up by connecting it to EV<sub>DD0</sub> or EV<sub>DD1</sub> via an external resistor, and be sure to keep inputting the V<sub>DD</sub> level to the TOOL0 pin before reset is released (pulling down this pin is prohibited).

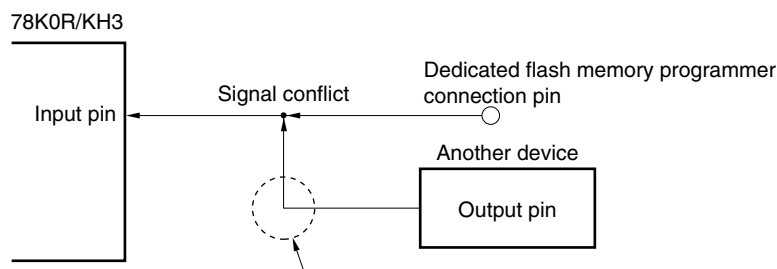
**Remark** The SAU and IIC0 pins are not used for communication between the 78K0R/KH3 and dedicated flash memory programmer, because single-line UART is used.

**25.4.3 RESET pin**

Signal conflict will occur if the reset signal of the dedicated flash memory programmer is connected to the  $\overline{\text{RESET}}$  pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.

The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer.

**Figure 25-5. Signal Conflict ( $\overline{\text{RESET}}$  Pin)**



In the flash memory programming mode, a signal output by another device will conflict with the signal output by the dedicated flash memory programmer. Therefore, isolate the signal of another device.

### 25.4.4 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to  $V_{DD}$  or  $V_{SS}$  via a resistor.

### 25.4.5 REGC pin

Connect the REGC pin to GND via a capacitor (0.47 to 1  $\mu F$ ) in the same manner as during normal operation. However, when using the STOP mode that has been entered since operation of the internal high-speed oscillation clock and external main system clock, 0.47  $\mu F$  is recommended. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

### 25.4.6 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

**Remark** In the flash memory programming mode, the internal high-speed oscillation clock ( $f_{IH}$ ) is used.

### 25.4.7 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 using the on-board supply voltage, be sure to connect the  $V_{DD}$  and  $V_{SS}$  pins to  $V_{DD}$  and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

Supply the same other power supplies ( $EV_{DD0}$ ,  $EV_{DD1}$ ,  $EV_{SS0}$ ,  $EV_{SS1}$ ,  $AV_{REF0}$ ,  $AV_{REF1}$ , and  $AV_{SS}$ ) as those in the normal operation mode.

## 25.5 Registers that Control Flash Memory

### (1) Background event control register (BECTL)

Even if the FLMD0 pin is not controlled externally, it can be controlled by software with the BECTL register to set the self-programming mode.

However, depending on the processing of the FLMD0 pin, it may not be possible to set the self-programming mode by software. When using BECTL, leaving the FLMD0 pin open is recommended. When pulling it down externally, use a resistor with a resistance of 100 k $\Omega$  or more. In addition, in the normal operation mode, use BECTL with the pull down selection. In the self-programming mode, the setting is switched to pull up in the self-programming library.

The BECTL register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

**Figure 25-6. Format of Background Event Control Register (BECTL)**

Address: FFFBEH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
BECTL	FLMDPUP	0	0	0	0	0	0	0

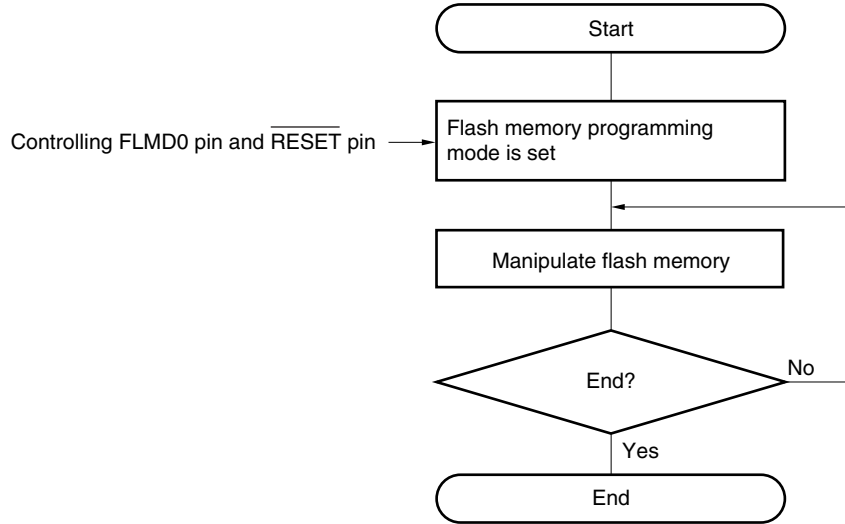
FLMDPUP	Software control of FLMD0 pin
0	Selects pull-down
1	Selects pull-up

## 25.6 Programming Method

### 25.6.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

**Figure 25-7. Flash Memory Manipulation Procedure**

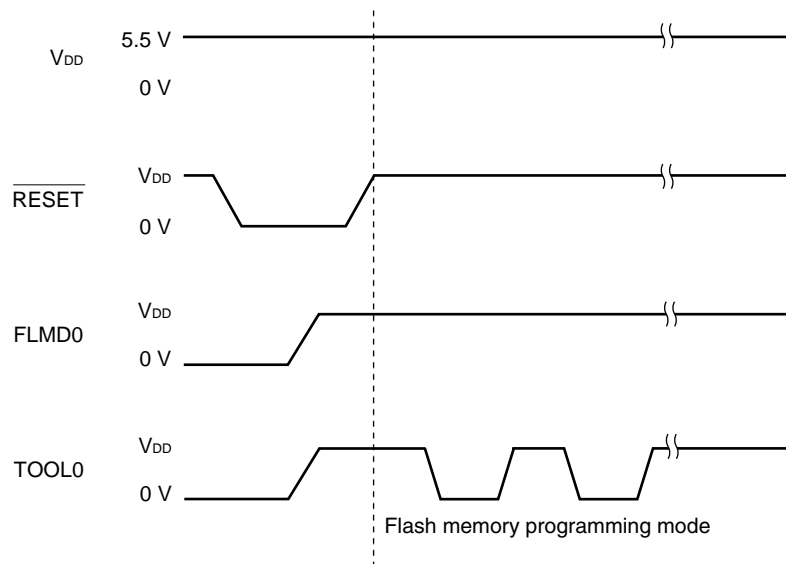


### 25.6.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the 78K0R/KH3 in the flash memory programming mode. To set the mode, set the FLMD0 pin and TOOL0 pin to  $V_{DD}$  and clear the reset signal.

Change the mode by using a jumper when writing the flash memory on-board.

**Figure 25-8. Flash Memory Programming Mode**



**Table 25-3. Relationship Between FLMD0 Pin and Operation Mode After Reset Release**

FLMD0	Operation Mode
0	Normal operation mode
V <sub>DD</sub>	Flash memory programming mode

**25.6.3 Selecting communication mode**

Communication mode of the 78K0R/KH3 as follows.

**Table 25-4. Communication Modes**

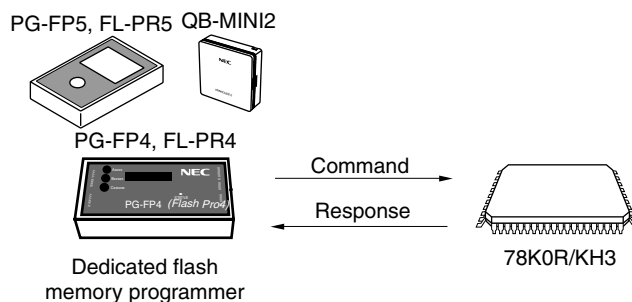
Communication Mode	Standard Setting <sup>Note 1</sup>				Pins Used
	Port	Speed <sup>Note 2</sup>	Frequency	Multiply Rate	
1-line mode (single-line UART)	UART	115,200 bps, 250,000 bps, 500,000 bps, 1 Mbps	–	–	TOOL0

- Notes**
1. Selection items for Standard settings on GUI of the flash memory programmer.
  2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

**25.6.4 Communication commands**

The 78K0R/KH3 communicates with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0R/KH3 are called commands, and the signals sent from the 78K0R/KH3 to the dedicated flash memory programmer are called response.

**Figure 25-9. Communication Commands**



The flash memory control commands of the 78K0R/KH3 are listed in the table below. All these commands are issued from the programmer and the 78K0R/KH3 perform processing corresponding to the respective commands.

**Table 25-5. Flash Memory Control Commands**

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Chip Erase	Erases the entire flash memory.
	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Write	Programming	Writes data to a specified area in the flash memory.
Getting information	Silicon Signature	Gets 78K0R/KH3 information (such as the part number and flash memory configuration).
	Version Get	Gets the 78K0R/KH3 firmware version.
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

The 78K0R/KH3 return a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0R/KH3 are listed below.

**Table 25-6. Response Names**

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

## 25.7 Security Settings

The 78K0R/KH3 supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the Security Set command. The security setting is valid when the programming mode is set next.

- Disabling batch erase (chip erase)

Execution of the block erase and batch erase (chip erase) commands for entire blocks in the flash memory is prohibited by this setting during on-board/off-board programming. Once execution of the batch erase (chip erase) command is prohibited, all of the prohibition settings (including prohibition of batch erase (chip erase)) can no longer be cancelled.

**Caution** After the security setting for the batch erase is set, erasure cannot be performed for the device. In addition, even if a write command is executed, data different from that which has already been written to the flash memory cannot be written, because the erase command is disabled.

- Disabling block erase

Execution of the block erase command for a specific block in the flash memory is prohibited during on-board/off-board programming. However, blocks can be erased by means of self programming.

- Disabling write

Execution of the write and block erase commands for entire blocks in the flash memory is prohibited during on-board/off-board programming. However, blocks can be written by means of self programming.

- Disabling rewriting boot cluster 0

Execution of the write command, block erase command, and batch erase (chip erase) command for boot cluster 0 (00000H to 00FFFH) in the flash memory is prohibited by this setting.

The batch erase (chip erase), block erase, write commands, and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by on-board/off-board programming and self programming. Each security setting can be used in combination.

All the security settings are cleared by executing the batch erase (chip erase) command.

Table 25-7 shows the relationship between the erase and write commands when the 78K0R/KH3 security function is enabled.

**Remark** To prohibit writing and erasing during self-programming, use the flash sealed window function (see 25.9.2 for detail).



Table 25-7. Relationship Between Enabling Security Function and Command

## (1) During on-board/off-board programming

Valid Security	Executed Command		
	Batch Erase (Chip Erase)	Block Erase	Write
Prohibition of batch erase (chip erase)	Cannot be erased in batch	Blocks cannot be erased.	Can be performed <sup>Note</sup> .
Prohibition of block erase	Can be erased in batch.		Can be performed.
Prohibition of writing			Cannot be performed.
Prohibition of rewriting boot cluster 0	Cannot be erased in batch	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

**Note** Confirm that no data has been written to the write area. Because data cannot be erased after batch erase (chip erase) is prohibited, do not write data if the data has not been erased.

## (2) During self programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of batch erase (chip erase)	Blocks can be erased.	Can be performed.
Prohibition of block erase		
Prohibition of writing		
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

**Remark** To prohibit writing and erasing during self-programming, use the flash sealed window function (see 25.9.2 for detail).

Table 25-8. Setting Security in Each Programming Mode

## (1) On-board/off-board programming

Security	Security Setting	How to Disable Security Setting
Prohibition of batch erase (chip erase)	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command
Prohibition of writing		
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

## (2) Self programming

Security	Security Setting	How to Disable Security Setting
Prohibition of batch erase (chip erase)	Set by using information library.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command during on-board/off-board programming (cannot be disabled during self programming)
Prohibition of writing		
Prohibition of rewriting boot cluster 0		

### 25.8 Processing Time of Each Command When Using PG-FP4 or PG-FP5 (Reference Values)

The processing time of each command (reference values) when using PG-FP4 or PG-FP5 as the dedicated flash memory programmer is shown below.

**Table 25-9. Processing Time of Each Command When Using PG-FP4 (Reference Values)**

PG-FP4 Command	Port: UART									
	Speed: 115200 bps					Speed: 1 Mbps				
	$\mu$ PD78F1174, $\mu$ PD78F1174A	$\mu$ PD78F1175, $\mu$ PD78F1175A	$\mu$ PD78F1176, $\mu$ PD78F1176A	$\mu$ PD78F1177, $\mu$ PD78F1177A	$\mu$ PD78F1178, $\mu$ PD78F1178A	$\mu$ PD78F1174, $\mu$ PD78F1174A	$\mu$ PD78F1175, $\mu$ PD78F1175A	$\mu$ PD78F1176, $\mu$ PD78F1176A	$\mu$ PD78F1177, $\mu$ PD78F1177A	$\mu$ PD78F1178, $\mu$ PD78F1178A
Signature	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	1 s (TYP.)	1 s (TYP.)
Blankcheck	1 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	2 s (TYP.)	2.5 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1.5 s (TYP.)	2 s (TYP.)	2.5 s (TYP.)
Erase	1 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	2 s (TYP.)	2.5 s (TYP.)	1 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	2 s (TYP.)	2.5 s (TYP.)
Program	19 s (TYP.)	26.5 s (TYP.)	35 s (TYP.)	51.5 s (TYP.)	68.5 s (TYP.)	6.5 s (TYP.)	9 s (TYP.)	12 s (TYP.)	17.5 s (TYP.)	23 s (TYP.)
Verify	16 s (TYP.)	23.5 s (TYP.)	31 s (TYP.)	46 s (TYP.)	61 s (TYP.)	4.5 s (TYP.)	6 s (TYP.)	8 s (TYP.)	11.5 s (TYP.)	15.5 s (TYP.)
E.P.V	20 s (TYP.)	28 s (TYP.)	36.5 s (TYP.)	53.5 s (TYP.)	71 s (TYP.)	7.5 s (TYP.)	10.5 s (TYP.)	13.5 s (TYP.)	19.5 s (TYP.)	25.5 s (TYP.)
Checksum	1 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	2 s (TYP.)	2.5 s (TYP.)	1 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	2 s (TYP.)	2.5 s (TYP.)
Security	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	1 s (TYP.)	1 s (TYP.)

**Table 25-10. Processing Time of Each Command When Using PG-FP5 (Reference Values)**

PG-FP5 Command	Port: UART									
	Speed: 115200 bps					Speed: 1 Mbps				
	$\mu$ PD78F1174, $\mu$ PD78F1174A	$\mu$ PD78F1175, $\mu$ PD78F1175A	$\mu$ PD78F1176, $\mu$ PD78F1176A	$\mu$ PD78F1177, $\mu$ PD78F1177A	$\mu$ PD78F1178, $\mu$ PD78F1178A	$\mu$ PD78F1174, $\mu$ PD78F1174A	$\mu$ PD78F1175, $\mu$ PD78F1175A	$\mu$ PD78F1176, $\mu$ PD78F1176A	$\mu$ PD78F1177, $\mu$ PD78F1177A	$\mu$ PD78F1178, $\mu$ PD78F1178A
Signature read	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)
Blank check	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	2 s (TYP.)	2.5 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1 s (TYP.)	2 s (TYP.)	2.5 s (TYP.)
Erase	1 s (TYP.)	1 s (TYP.)	1.5 s (TYP.)	2 s (TYP.)	2.5 s (TYP.)	1 s (TYP.)	1 s (TYP.)	1.5 s (TYP.)	2 s (TYP.)	2.5 s (TYP.)
Program	17.5 s (TYP.)	26 s (TYP.)	34 s (TYP.)	51 s (TYP.)	67.5 s (TYP.)	6 s (TYP.)	8.5 s (TYP.)	11 s (TYP.)	16.5 s (TYP.)	22 s (TYP.)
Verify	15.5 s (TYP.)	23 s (TYP.)	30.5 s (TYP.)	45.5 s (TYP.)	60 s (TYP.)	4 s (TYP.)	5.5 s (TYP.)	7.5 s (TYP.)	11 s (TYP.)	14 s (TYP.)
Auto-procedure	18 s (TYP.)	26.5 s (TYP.)	35 s (TYP.)	52 s (TYP.)	69 s (TYP.)	6 s (TYP.)	9 s (TYP.)	12 s (TYP.)	18 s (TYP.)	23.5 s (TYP.)
Checksum	1 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	2 s (TYP.)	2.5 s (TYP.)	1 s (TYP.)	1.5 s (TYP.)	1.5 s (TYP.)	2 s (TYP.)	2.5 s (TYP.)
Security	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)	0.5 s (TYP.)

## 25.9 Flash Memory Programming by Self-Programming

The 78K0R/KH3 supports a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using the 78K0R/KH3 self-programming library, it can be used to upgrade the program in the field.

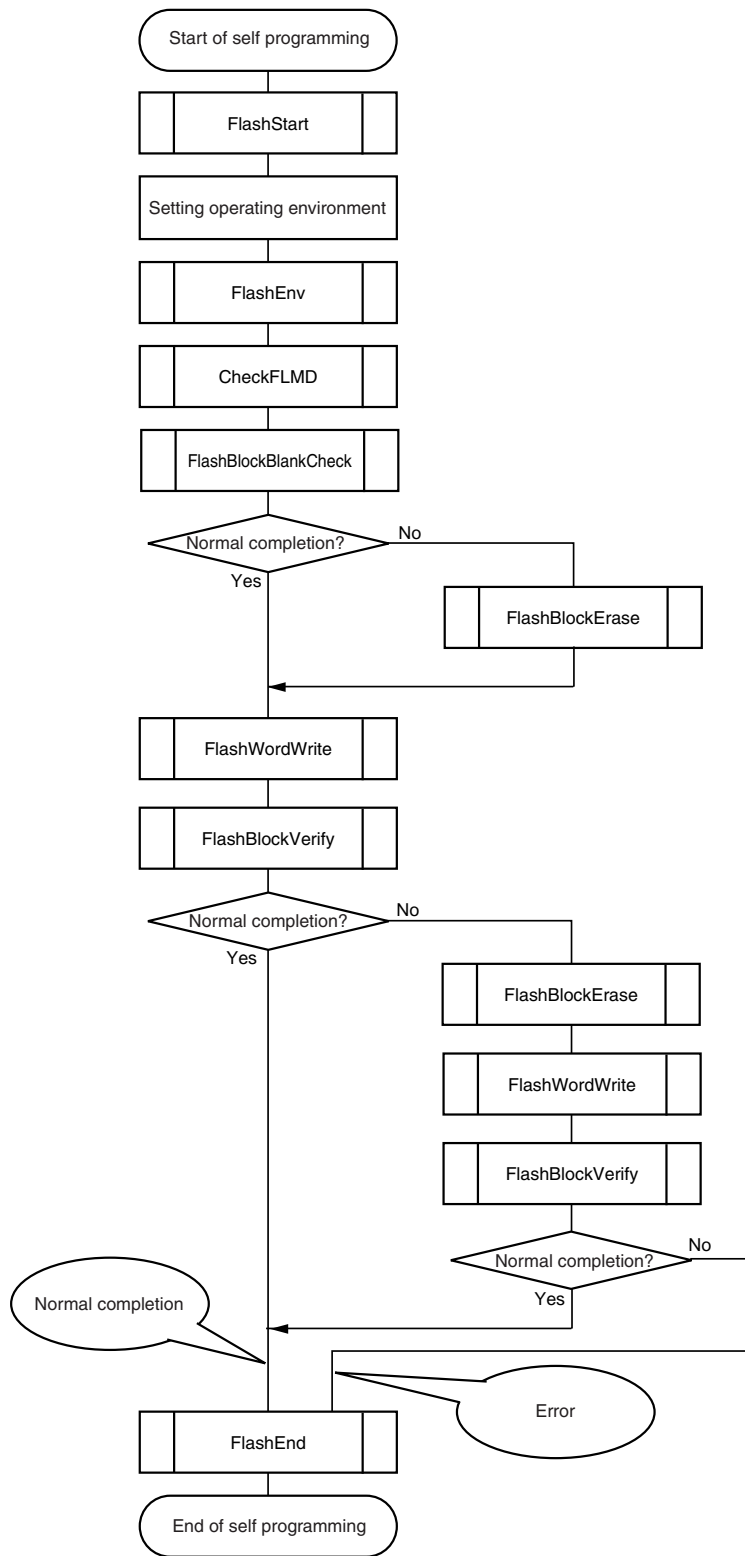
If an interrupt occurs during self-programming, self-programming can be temporarily stopped and interrupt servicing can be executed. If an unmasked interrupt request is generated in the EI state, the request branches directly from the self-programming library to the interrupt routine. After the self-programming mode is later restored, self-programming can be resumed. However, the interrupt response time is different from that of the normal operation mode.

**Remark** For details of the self-programming function and the 78K0R/KH3 self-programming library, refer to **78K0R Microcontroller Self Programming Library Type01 User's Manual (U18706E)**.

- Cautions**
1. The self-programming function cannot be used when the CPU operates with the subsystem clock.
  2. In the self-programming mode, call the self-programming start library (FlashStart).
  3. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the self-programming library.
  4. The self-programming function is disabled in the low consumption current mode. For details of the low consumption current mode, see CHAPTER 23 REGULATOR.
  5. Disable DMA operation (DENn = 0) during the execution of self programming library functions.

The following figure illustrates a flow of rewriting the flash memory by using a self programming library.

**Figure 25-10. Flow of Self Programming (Rewriting Flash Memory)**



**Remark** For details of the self programming library, refer to **78K0R Microcontroller Self Programming Library Type01 User’s Manual (U18706E)**.

### 25.9.1 Boot swap function

If rewriting the boot area failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.

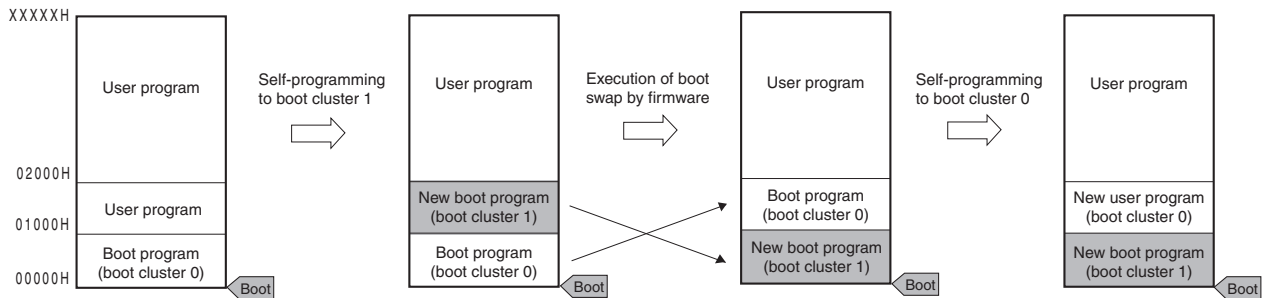
The boot swap function is used to avoid this problem.

Before erasing boot cluster 0<sup>Note</sup>, which is a boot program area, by self-programming, write a new boot program to boot cluster 1 in advance. When the program has been correctly written to boot cluster 1, swap this boot cluster 1 and boot cluster 0 by using the set information function of the firmware of the 78K0R/KH3, so that boot cluster 1 is used as a boot area. After that, erase or write the original boot program area, boot cluster 0.

As a result, even if a power failure occurs while the boot programming area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 to be swapped when the program is reset and started next.

**Note** A boot cluster is a 4 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

**Figure 25-11. Boot Swap Function**

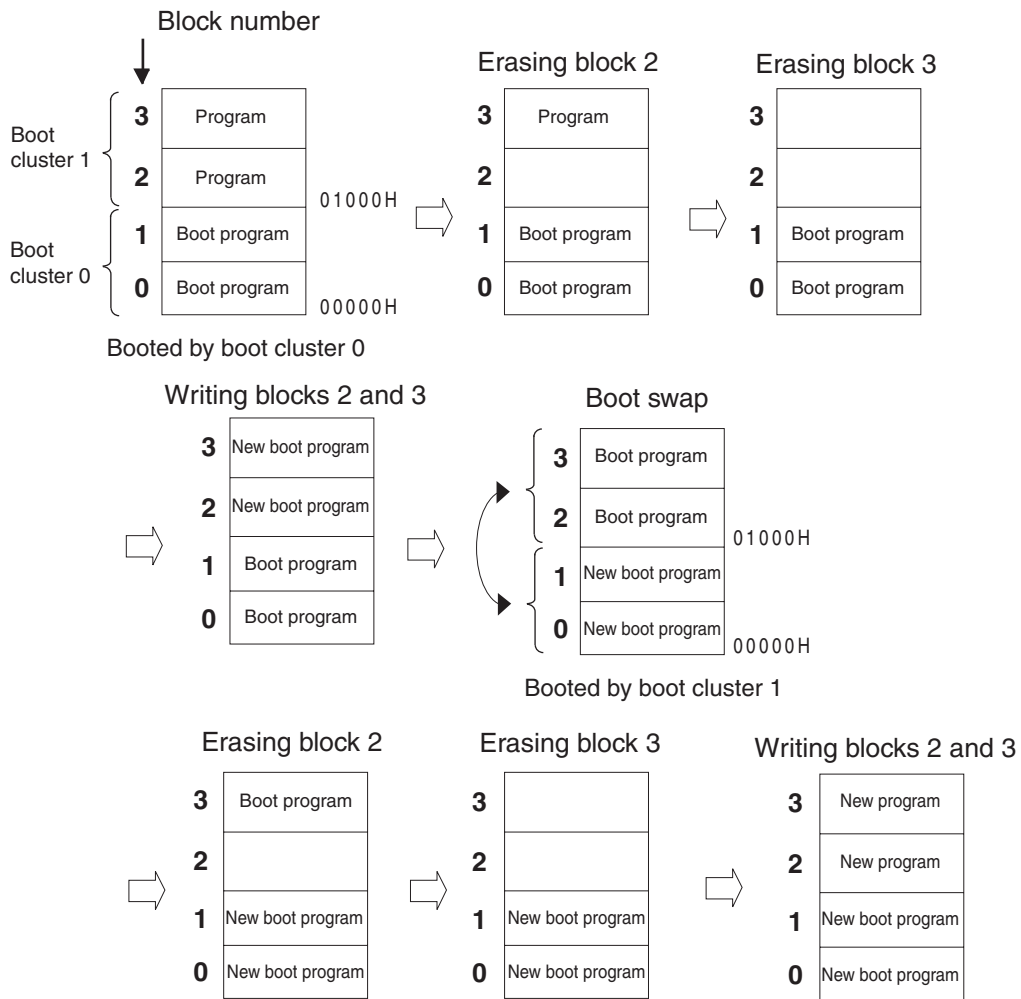


In an example of above figure, it is as follows.

Boot cluster 0: Boot program area before boot swap

Boot cluster 1: Boot program area after boot swap

Figure 25-12. Example of Executing Boot Swapping



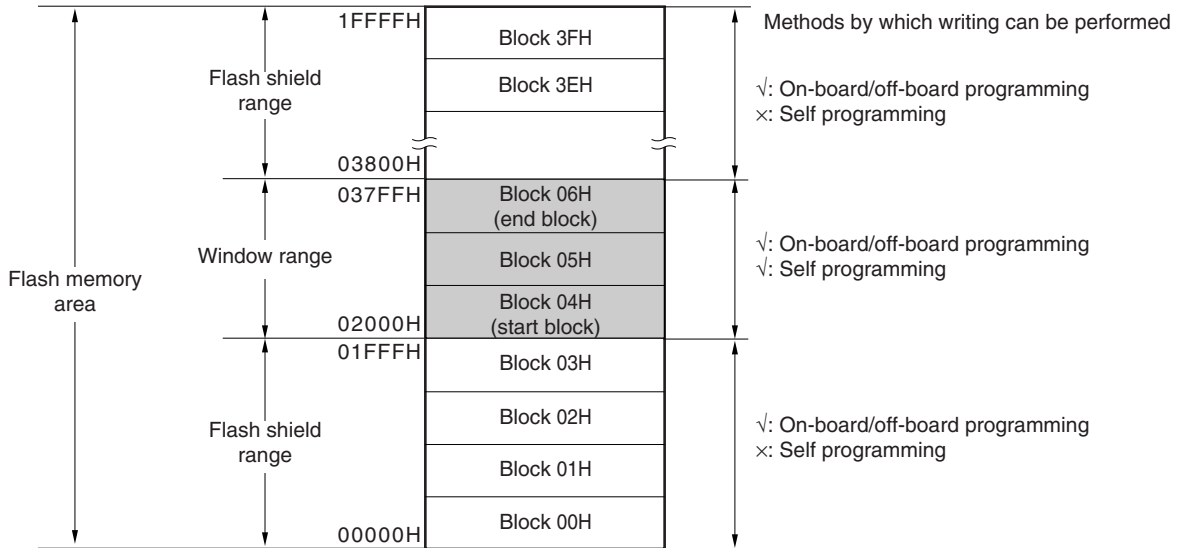
**25.9.2 Flash shield window function**

The flash shield window function is provided as one of the security functions for self programming. It disables writing to and erasing areas outside the range specified as a window only during self programming.

The window range can be set by specifying the start and end blocks. The window range can be set or changed during both on-board/off-board programming and self programming.

Writing to and erasing areas outside the window range are disabled during self programming. During on-board/off-board programming, however, areas outside the range specified as a window can be written and erased.

**Figure 25-13. Flash Shield Window Setting Example**  
 (Target Devices:  $\mu$ PD78F1174, 78F1174A, Start Block: 04H, End Block: 06H)



**Caution** If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.

**Table 25-11. Relationship Between Flash Shield Window Function Setting/Change Methods and Commands**

Programming Conditions	Window Range Setting/Change Methods	Execution Commands	
		Block Erase	Write
Self-programming	Specify the starting and ending blocks by the set information library.	Block erasing is enabled only within the window range.	Writing is enabled only within the range of window range.
On-board/off-board programming	Specify the starting and ending blocks on GUI of dedicated flash memory programmer, etc.	Block erasing is enabled also outside the window range.	Writing is enabled also outside the window range.

**Remark** See 25.7 Security Settings to prohibit writing/erasing during on-board/off-board programming.

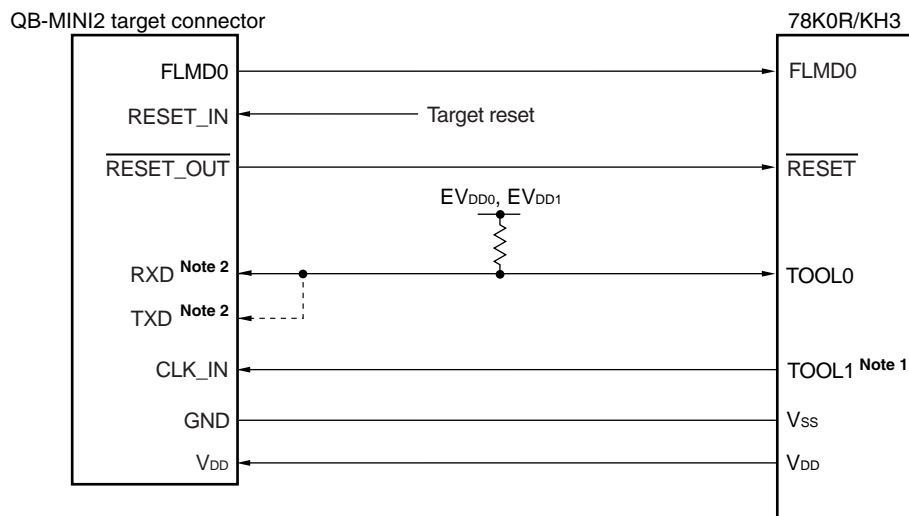
## CHAPTER 26 ON-CHIP DEBUG FUNCTION

### 26.1 Connecting QB-MINI2 to 78K0R/KH3

The 78K0R/KH3 uses the  $V_{DD}$ , FLMD0,  $\overline{\text{RESET}}$ , TOOL0, TOOL1<sup>Note</sup>, and  $V_{SS}$  pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2).

**Caution** The 78K0R/KH3 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. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 26-1. Connection Example of QB-MINI2 and 78K0R/KH3



- Notes**
1. Connection is not required for communication in 1-line mode but required for communication in 2-line mode. At this time, perform necessary connections according to Table 2-2 Connection of Unused Pins since TOOL1 is an unused pin when QB-MINI2 is unconnected.
  2. Connecting the dotted line is not necessary since RXD and TXD are shorted within QB-MINI2. When using the other flash memory programmer, RXD and TXD may not be shorted within the programmer. In this case, they must be shorted on the target system.

**Caution** When communicating in 2-line mode, a clock with a frequency of half that of the CPU clock frequency is output from the TOOL1 pin. A resistor or ferrite bead can be used as a countermeasure against fluctuation of the power supply caused by that clock.

**Remark** The FLMD0 pin is recommended to be open for self-programming in on-chip debugging. To pull down externally, use a resistor of 100 k $\Omega$  or more.

1-line mode (single line UART) using the TOOL0 pin or 2-line mode using the TOOL0 and TOOL1 pins is used for serial communication. For flash memory programming, 1-line mode is used. 1-line mode or 2-line mode is used for on-chip debugging. Table 26-1 lists the differences between 1-line mode and 2-line mode.



**Table 26-1. Differences Between 1-Line Mode and 2-Line Mode**

Communication Mode	Flash Memory Programming Function	Debugging Function
1-line mode	Available	<ul style="list-style-type: none"> <li>• Pseudo real-time RAM monitor (RRM) function not supported</li> </ul>
2-line mode	None	<ul style="list-style-type: none"> <li>• Pseudo real-time RAM monitor (RRM) function supported</li> </ul>

**Remark** 2-line mode is not used for flash programming, however, even if TOOL1 pin is connected with CLK\_IN of QB-MINI2, writing is performed normally with no problem.

## 26.2 On-Chip Debug Security ID

The 78K0R/KH3 has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 24 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

When the boot swap function is used, also set a value that is the same as that of 010C3H and 010C4H to 010CDH in advance, because 000C3H, 000C4H to 000CDH and 010C3H, and 010C4H to 010CDH are switched.

For details on the on-chip debug security ID, refer to the **QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual (U18371E)**.

**Table 26-2. On-Chip Debug Security ID**

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
010C4H to 010CDH	

## 26.3 Securing of user resources

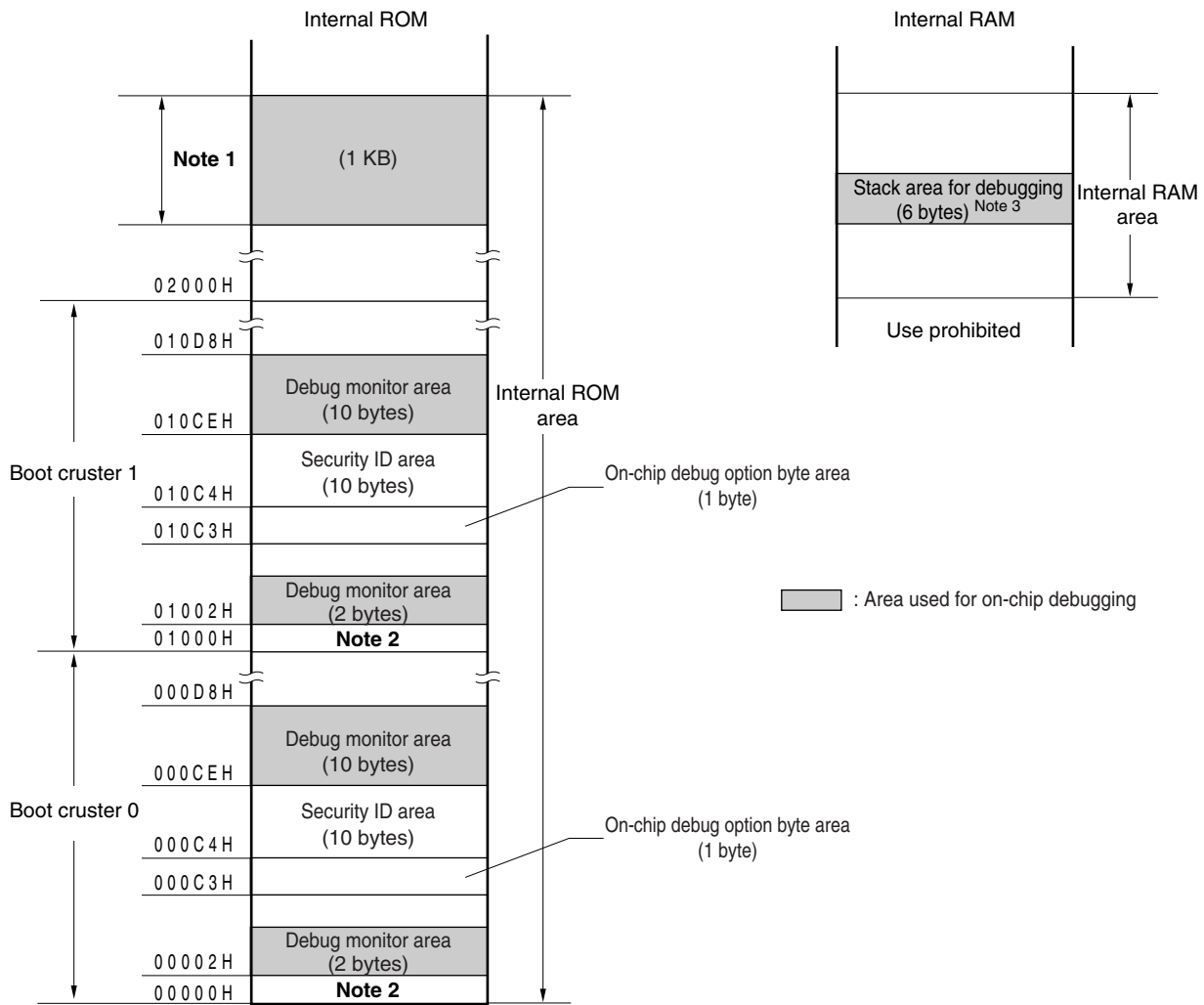
To perform communication between the 78K0R/KH3 and QB-MINI2, as well as each debug function, the securing of memory space must be done beforehand.

If NEC Electronics assembler RA78K0R or compiler CC78K0R is used, the items can be set by using linker options.

### (1) Securement of memory space

The shaded portions in Figure 26-2 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 26-2. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on products as follows.

Products	Internal ROM	Address
μPD78F1174, 78F1174A	128 KB	1FC00H-1FFFFH
μPD78F1175, 78F1175A	192 KB	2FC00H-2FFFFH
μPD78F1176, 78F1176A	256 KB	3FC00H-3FFFFH
μPD78F1177, 78F1177A	384 KB	5FC00H-5FFFFH
μPD78F1178, 78F1178A	512 KB	7FC00H-7FFFFH

- In debugging, reset vector is rewritten to address allocated to a monitor program.
- 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, 6 extra bytes are consumed for the stack area used.

For details of the way to secure of the memory space, refer to the **QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual (U18371E)**.

## CHAPTER 27 BCD CORRECTION CIRCUIT

### 27.1 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 BCDADJ register.

### 27.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

#### (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.

BCDADJ is read by an 8-bit memory manipulation instruction.

Reset input sets this register to undefined.

**Figure 27-1. Format of BCD Correction Result Register (BCDADJ)**

Address: F00FEH    After reset: undefined    R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

## 27.3 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 BCDADJ register.
- <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:  $85 + 15 = 100$

<R>

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 BCDADJ register.
- <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 28 INSTRUCTION SET

This chapter lists the instructions in the 78K0R microcontroller instruction set. For details of each operation and operation code, refer to the separate document **78K0R Microcontrollers Instructions User's Manual (U17792E)**.

**Remark** The shaded parts of the tables in **Table 28-5 Operation List** indicate the operation or instruction format that is newly added for the 78K0R microcontrollers.

## 28.1 Conventions Used in Operation List

### 28.1.1 Operand identifiers and specification methods

Operands are described in the “Operand” column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Alphabetic letters in capitals and the symbols, #, !, !!, \$, \$!, [ ], and ES: are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: 16-bit absolute address specification
- !!: 20-bit absolute address specification
- \$: 8-bit relative address specification
- \$!: 16-bit relative address specification
- [ ]: Indirect address specification
- ES:: Extension address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, !!, \$, \$!, [ ], and ES: symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

**Table 28-1. Operand Identifiers and Specification Methods**

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol (SFR symbol) FFF00H to FFFFFH
sfrp	Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only <sup>Note</sup> ) FFF00H to FFFFFH
saddr	FFE20H to FFF1FH Immediate data or labels
saddrp	FFE20H to FF1FH Immediate data or labels (even addresses only <sup>Note</sup> )
addr20	00000H to FFFFFH Immediate data or labels
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions <sup>Note</sup> )
addr5	0080H to 00BFH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

**Note** Bit 0 = 0 when an odd address is specified.

**Remark** The special function registers can be described to operand sfr as symbols. See **Table 3-5 SFR List** for the symbols of the special function registers.

The extended special function registers can be described to operand !addr16 as symbols. See **Table 3-6 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

## 28.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 28-2. Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X <sub>H</sub> , X <sub>L</sub>	16-bit registers: X <sub>H</sub> = higher 8 bits, X <sub>L</sub> = lower 8 bits
X <sub>S</sub> , X <sub>H</sub> , X <sub>L</sub>	20-bit registers: X <sub>S</sub> = (bits 19 to 16), X <sub>H</sub> = (bits 15 to 8), X <sub>L</sub> = (bits 7 to 0)
^	Logical product (AND)
∨	Logical sum (OR)
⊕	Exclusive logical sum (exclusive OR)
—	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)



**28.1.3 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 28-3. 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

**28.1.4 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.

<R> An interrupt and DMA transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

**Table 28-4. 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.

## 28.2 Operation List

Table 28-5. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	–	r ← byte			
		saddr, #byte	3	1	–	(saddr) ← byte			
		sfr, #byte	3	1	–	sfr ← byte			
		laddr16, #byte	4	1	–	(addr16) ← byte			
		A, r <sup>Note 3</sup>	1	1	–	A ← r			
		r, A <sup>Note 3</sup>	1	1	–	r ← A			
		A, saddr	2	1	–	A ← (saddr)			
		saddr, A	2	1	–	(saddr) ← A			
		A, sfr	2	1	–	A ← sfr			
		sfr, A	2	1	–	sfr ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		!addr16, A	3	1	–	(addr16) ← A			
		PSW, #byte	3	3	–	PSW ← byte	x	x	x
		A, PSW	2	1	–	A ← PSW			
		PSW, A	2	3	–	PSW ← A	x	x	x
		ES, #byte	2	1	–	ES ← byte			
		ES, saddr	3	1	–	ES ← (saddr)			
		A, ES	2	1	–	A ← ES			
		ES, A	2	1	–	ES ← A			
		CS, #byte	3	1	–	CS ← byte			
		A, CS	2	1	–	A ← CS			
		CS, A	2	1	–	CS ← A			
		A, [DE]	1	1	4	A ← (DE)			
		[DE], A	1	1	–	(DE) ← A			
		[DE + byte], #byte	3	1	–	(DE + byte) ← byte			
		A, [DE + byte]	2	1	4	A ← (DE + byte)			
		[DE + byte], A	2	1	–	(DE + byte) ← A			
		A, [HL]	1	1	4	A ← (HL)			
[HL], A	1	1	–	(HL) ← A					
[HL + byte], #byte	3	1	–	(HL + byte) ← byte					

**Notes** 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

**Remarks** 1. One instruction clock cycle is one cycle of the CPU clock (f<sub>CPU</sub>) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).

3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (2/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	A, [HL + byte]	2	1	4	$A \leftarrow (HL + \text{byte})$			
		[HL + byte], A	2	1	–	$(HL + \text{byte}) \leftarrow A$			
		A, [HL + B]	2	1	4	$A \leftarrow (HL + B)$			
		[HL + B], A	2	1	–	$(HL + B) \leftarrow A$			
		A, [HL + C]	2	1	4	$A \leftarrow (HL + C)$			
		[HL + C], A	2	1	–	$(HL + C) \leftarrow A$			
		word[B], #byte	4	1	–	$(B + \text{word}) \leftarrow \text{byte}$			
		A, word[B]	3	1	4	$A \leftarrow (B + \text{word})$			
		word[B], A	3	1	–	$(B + \text{word}) \leftarrow A$			
		word[C], #byte	4	1	–	$(C + \text{word}) \leftarrow \text{byte}$			
		A, word[C]	3	1	4	$A \leftarrow (C + \text{word})$			
		word[C], A	3	1	–	$(C + \text{word}) \leftarrow A$			
		word[BC], #byte	4	1	–	$(BC + \text{word}) \leftarrow \text{byte}$			
		A, word[BC]	3	1	4	$A \leftarrow (BC + \text{word})$			
		word[BC], A	3	1	–	$(BC + \text{word}) \leftarrow A$			
		[SP + byte], #byte	3	1	–	$(SP + \text{byte}) \leftarrow \text{byte}$			
		A, [SP + byte]	2	1	–	$A \leftarrow (SP + \text{byte})$			
		[SP + byte], A	2	1	–	$(SP + \text{byte}) \leftarrow A$			
		B, saddr	2	1	–	$B \leftarrow (\text{saddr})$			
		B, !addr16	3	1	4	$B \leftarrow (\text{addr16})$			
		C, saddr	2	1	–	$C \leftarrow (\text{saddr})$			
		C, !addr16	3	1	4	$C \leftarrow (\text{addr16})$			
		X, saddr	2	1	–	$X \leftarrow (\text{saddr})$			
		X, !addr16	3	1	4	$X \leftarrow (\text{addr16})$			
		ES:!addr16, #byte	5	2	–	$(ES, \text{addr16}) \leftarrow \text{byte}$			
		A, ES:!addr16	4	2	5	$A \leftarrow (ES, \text{addr16})$			
		ES:!addr16, A	4	2	–	$(ES, \text{addr16}) \leftarrow A$			
		A, ES:[DE]	2	2	5	$A \leftarrow (ES, DE)$			
		ES:[DE], A	2	2	–	$(ES, DE) \leftarrow A$			
		ES:[DE + byte], #byte	4	2	–	$((ES, DE) + \text{byte}) \leftarrow \text{byte}$			
A, ES:[DE + byte]	3	2	5	$A \leftarrow ((ES, DE) + \text{byte})$					
ES:[DE + byte], A	3	2	–	$((ES, DE) + \text{byte}) \leftarrow A$					

**Notes** 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

**Remarks** 1. One instruction clock cycle is one cycle of the CPU clock ( $f_{CPU}$ ) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).

3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (3/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	A, ES:[HL]	2	2	5	$A \leftarrow (ES, HL)$			
		ES:[HL], A	2	2	–	$(ES, HL) \leftarrow A$			
		ES:[HL + byte], #byte	4	2	–	$((ES, HL) + \text{byte}) \leftarrow \text{byte}$			
		A, ES:[HL + byte]	3	2	5	$A \leftarrow ((ES, HL) + \text{byte})$			
		ES:[HL + byte], A	3	2	–	$((ES, HL) + \text{byte}) \leftarrow A$			
		A, ES:[HL + B]	3	2	5	$A \leftarrow ((ES, HL) + B)$			
		ES:[HL + B], A	3	2	–	$((ES, HL) + B) \leftarrow A$			
		A, ES:[HL + C]	3	2	5	$A \leftarrow ((ES, HL) + C)$			
		ES:[HL + C], A	3	2	–	$((ES, HL) + C) \leftarrow A$			
		ES:word[B], #byte	5	2	–	$((ES, B) + \text{word}) \leftarrow \text{byte}$			
		A, ES:word[B]	4	2	5	$A \leftarrow ((ES, B) + \text{word})$			
		ES:word[B], A	4	2	–	$((ES, B) + \text{word}) \leftarrow A$			
		ES:word[C], #byte	5	2	–	$((ES, C) + \text{word}) \leftarrow \text{byte}$			
		A, ES:word[C]	4	2	5	$A \leftarrow ((ES, C) + \text{word})$			
		ES:word[C], A	4	2	–	$((ES, C) + \text{word}) \leftarrow A$			
		ES:word[BC], #byte	5	2	–	$((ES, BC) + \text{word}) \leftarrow \text{byte}$			
		A, ES:word[BC]	4	2	5	$A \leftarrow ((ES, BC) + \text{word})$			
		ES:word[BC], A	4	2	–	$((ES, BC) + \text{word}) \leftarrow A$			
		B, ES:!addr16	4	2	5	$B \leftarrow (ES, \text{addr16})$			
		C, ES:!addr16	4	2	5	$C \leftarrow (ES, \text{addr16})$			
X, ES:!addr16	4	2	5	$X \leftarrow (ES, \text{addr16})$					
XCH	A, r	<sup>Note 3</sup> 1 (r = X) 2 (other than r = X)	1 (r = X) 2 (other than r = X)	1	–	$A \leftrightarrow r$			
		A, saddr	3	2	–	$A \leftrightarrow (\text{saddr})$			
		A, sfr	3	2	–	$A \leftrightarrow \text{sfr}$			
		A, !addr16	4	2	–	$A \leftrightarrow (\text{addr16})$			
		A, [DE]	2	2	–	$A \leftrightarrow (DE)$			
		A, [DE + byte]	3	2	–	$A \leftrightarrow (DE + \text{byte})$			
		A, [HL]	2	2	–	$A \leftrightarrow (HL)$			
		A, [HL + byte]	3	2	–	$A \leftrightarrow (HL + \text{byte})$			
		A, [HL + B]	2	2	–	$A \leftrightarrow (HL + B)$			
A, [HL + C]	2	2	–	$A \leftrightarrow (HL + C)$					

**Notes** 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

**Remarks** 1. One instruction clock cycle is one cycle of the CPU clock (f<sub>CPU</sub>) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).

3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (4/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag			
				Note 1	Note 2		Z	AC	CY	
8-bit data transfer	XCH	A, ES:!addr16	5	3	–	A ←→ (ES, addr16)				
		A, ES:[DE]	3	3	–	A ←→ (ES, DE)				
		A, ES:[DE + byte]	4	3	–	A ←→ ((ES, DE) + byte)				
		A, ES:[HL]	3	3	–	A ←→ (ES, HL)				
		A, ES:[HL + byte]	4	3	–	A ←→ ((ES, HL) + byte)				
		A, ES:[HL + B]	3	3	–	A ←→ ((ES, HL) + B)				
		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				
		saddr	2	1	–	(saddr) ← 01H				
		!addr16	3	1	–	(addr16) ← 01H				
		ES:!addr16	4	2	–	(ES, addr16) ← 01H				
	CLRB	A	1	1	–	A ← 00H				
		X	1	1	–	X ← 00H				
		B	1	1	–	B ← 00H				
		C	1	1	–	C ← 00H				
		saddr	2	1	–	(saddr) ← 00H				
		!addr16	3	1	–	(addr16) ← 00H				
		ES:!addr16	4	2	–	(ES,addr16) ← 00H				
	MOVS	[HL + byte], X	3	1	–	(HL + byte) ← X	×		×	
		ES:[HL + byte], X	4	2	–	(ES, HL + byte) ← X	×		×	
	16-bit data transfer	MOVW	rp, #word	3	1	–	rp ← word			
			saddrp, #word	4	1	–	(saddrp) ← word			
			sfrp, #word	4	1	–	sfrp ← word			
			AX, saddrp	2	1	–	AX ← (saddrp)			
			saddrp, AX	2	1	–	(saddrp) ← AX			
AX, sfrp			2	1	–	AX ← sfrp				
sfrp, AX			2	1	–	sfrp ← AX				
AX, rp			Note 3	1	1	–	AX ← rp			
rp, AX			Note 3	1	1	–	rp ← AX			

**Notes** 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except rp = AX

**Remarks** 1. One instruction clock cycle is one cycle of the CPU clock (f<sub>cpu</sub>) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).

3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (5/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, !addr16	3	1	4	AX ← (addr16)			
		!addr16, AX	3	1	–	(addr16) ← AX			
		AX, [DE]	1	1	4	AX ← (DE)			
		[DE], AX	1	1	–	(DE) ← AX			
		AX, [DE + byte]	2	1	4	AX ← (DE + byte)			
		[DE + byte], AX	2	1	–	(DE + byte) ← AX			
		AX, [HL]	1	1	4	AX ← (HL)			
		[HL], AX	1	1	–	(HL) ← AX			
		AX, [HL + byte]	2	1	4	AX ← (HL + byte)			
		[HL + byte], AX	2	1	–	(HL + byte) ← AX			
		AX, word[B]	3	1	4	AX ← (B + word)			
		word[B], AX	3	1	–	(B + word) ← AX			
		AX, word[C]	3	1	4	AX ← (C + word)			
		word[C], AX	3	1	–	(C + word) ← AX			
		AX, word[BC]	3	1	4	AX ← (BC + word)			
		word[BC], AX	3	1	–	(BC + word) ← AX			
		AX, [SP + byte]	2	1	–	AX ← (SP + byte)			
		[SP + byte], AX	2	1	–	(SP + byte) ← AX			
		BC, saddrp	2	1	–	BC ← (saddrp)			
		BC, !addr16	3	1	4	BC ← (addr16)			
		DE, saddrp	2	1	–	DE ← (saddrp)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		HL, saddrp	2	1	–	HL ← (saddrp)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)			
		ES:!addr16, AX	4	2	–	(ES, addr16) ← AX			
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)			
		ES:[DE], AX	2	2	–	(ES, DE) ← AX			
		AX, ES:[DE + byte]	3	2	5	AX ← ((ES, DE) + byte)			
		ES:[DE + byte], AX	3	2	–	((ES, DE) + byte) ← AX			
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)			
		ES:[HL], AX	2	2	–	(ES, HL) ← AX			

- Notes**
1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.
  2. When the program memory area is accessed.

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock ( $f_{CPU}$ ) selected by the system clock control register (CKC).
  2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).
  3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (6/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, ES:[HL + byte]	3	2	5	AX ← ((ES, HL) + byte)			
		ES:[HL + byte], AX	3	2	–	((ES, HL) + byte) ← AX			
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	–	((ES, B) + word) ← AX			
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	–	((ES, C) + word) ← AX			
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	–	((ES, BC) + word) ← AX			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
	HL, ES:!addr16	4	2	5	HL ← (ES, addr16)				
	XCHW	AX, rp <sup>Note 3</sup>	1	1	–	AX ↔ rp			
	ONEW	AX	1	1	–	AX ← 0001H			
		BC	1	1	–	BC ← 0001H			
CLRW	AX	1	1	–	AX ← 0000H				
	BC	1	1	–	BC ← 0000H				
8-bit operation	ADD	A, #byte	2	1	–	A, CY ← A + byte	x	x	x
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) + byte	x	x	x
		A, r <sup>Note 4</sup>	2	1	–	A, CY ← A + r	x	x	x
		r, A	2	1	–	r, CY ← r + A	x	x	x
		A, saddr	2	1	–	A, CY ← A + (saddr)	x	x	x
		A, !addr16	3	1	4	A, CY ← A + (addr16)	x	x	x
		A, [HL]	1	1	4	A, CY ← A + (HL)	x	x	x
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte)	x	x	x
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B)	x	x	x
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C)	x	x	x
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	x	x	x
		A, ES:[HL]	2	2	5	A, CY ← A + (ES, HL)	x	x	x
		A, ES:[HL + byte]	3	2	5	A, CY ← A + ((ES, HL) + byte)	x	x	x
		A, ES:[HL + B]	3	2	5	A, CY ← A + ((ES, HL) + B)	x	x	x
A, ES:[HL + C]	3	2	5	A, CY ← A + ((ES, HL) + C)	x	x	x		

- Notes**
1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.
  2. When the program memory area is accessed.
  3. Except rp = AX
  4. Except r = A

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock (f<sub>cpu</sub>) selected by the system clock control register (CKC).
  2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).
  3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (7/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	ADDC	A, #byte	2	1	–	A, CY ← A + byte + CY	x	x	x
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) + byte + CY	x	x	x
		A, r <sup>Note 3</sup>	2	1	–	A, CY ← A + r + CY	x	x	x
		r, A	2	1	–	r, CY ← r + A + CY	x	x	x
		A, saddr	2	1	–	A, CY ← A + (saddr) + CY	x	x	x
		A, !addr16	3	1	4	A, CY ← A + (addr16) + CY	x	x	x
		A, [HL]	1	1	4	A, CY ← A + (HL) + CY	x	x	x
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte) + CY	x	x	x
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B) + CY	x	x	x
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C) + CY	x	x	x
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16) + CY	x	x	x
		A, ES:[HL]	2	2	5	A, CY ← A + (ES, HL) + CY	x	x	x
		A, ES:[HL + byte]	3	2	5	A, CY ← A + ((ES, HL) + byte) + CY	x	x	x
		A, ES:[HL + B]	3	2	5	A, CY ← A + ((ES, HL) + B) + CY	x	x	x
		A, ES:[HL + C]	3	2	5	A, CY ← A + ((ES, HL) + C) + CY	x	x	x
	SUB	A, #byte	2	1	–	A, CY ← A – byte	x	x	x
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte	x	x	x
		A, r <sup>Note 3</sup>	2	1	–	A, CY ← A – r	x	x	x
		r, A	2	1	–	r, CY ← r – A	x	x	x
		A, saddr	2	1	–	A, CY ← A – (saddr)	x	x	x
		A, !addr16	3	1	4	A, CY ← A – (addr16)	x	x	x
		A, [HL]	1	1	4	A, CY ← A – (HL)	x	x	x
		A, [HL + byte]	2	1	4	A, CY ← A – (HL + byte)	x	x	x
		A, [HL + B]	2	1	4	A, CY ← A – (HL + B)	x	x	x
		A, [HL + C]	2	1	4	A, CY ← A – (HL + C)	x	x	x
		A, ES:!addr16	4	2	5	A, CY ← A – (ES:addr16)	x	x	x
		A, ES:[HL]	2	2	5	A, CY ← A – (ES:HL)	x	x	x

- Notes**
1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.
  2. When the program memory area is accessed.
  3. Except r = A

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock ( $f_{CPU}$ ) selected by the system clock control register (CKC).
  2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).
  3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.



Table 28-5. Operation List (8/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	–	$A, CY \leftarrow A - \text{byte} - CY$	x	x	x
		saddr, #byte	3	2	–	$(\text{saddr}), CY \leftarrow (\text{saddr}) - \text{byte} - CY$	x	x	x
		A, r <sup>Note 3</sup>	2	1	–	$A, CY \leftarrow A - r - CY$	x	x	x
		r, A	2	1	–	$r, CY \leftarrow r - A - CY$	x	x	x
		A, saddr	2	1	–	$A, CY \leftarrow A - (\text{saddr}) - CY$	x	x	x
		A, !addr16	3	1	4	$A, CY \leftarrow A - (\text{addr16}) - CY$	x	x	x
		A, [HL]	1	1	4	$A, CY \leftarrow A - (\text{HL}) - CY$	x	x	x
		A, [HL + byte]	2	1	4	$A, CY \leftarrow A - (\text{HL} + \text{byte}) - CY$	x	x	x
		A, [HL + B]	2	1	4	$A, CY \leftarrow A - (\text{HL} + B) - CY$	x	x	x
		A, [HL + C]	2	1	4	$A, CY \leftarrow A - (\text{HL} + C) - CY$	x	x	x
		A, ES:!addr16	4	2	5	$A, CY \leftarrow A - (\text{ES:addr16}) - CY$	x	x	x
		A, ES:[HL]	2	2	5	$A, CY \leftarrow A - (\text{ES:HL}) - CY$	x	x	x
		A, ES:[HL + byte]	3	2	5	$A, CY \leftarrow A - ((\text{ES:HL}) + \text{byte}) - CY$	x	x	x
		A, ES:[HL + B]	3	2	5	$A, CY \leftarrow A - ((\text{ES:HL}) + B) - CY$	x	x	x
		A, ES:[HL + C]	3	2	5	$A, CY \leftarrow A - ((\text{ES:HL}) + C) - CY$	x	x	x
	AND	A, #byte	2	1	–	$A \leftarrow A \wedge \text{byte}$	x		
		saddr, #byte	3	2	–	$(\text{saddr}) \leftarrow (\text{saddr}) \wedge \text{byte}$	x		
		A, r <sup>Note 3</sup>	2	1	–	$A \leftarrow A \wedge r$	x		
		r, A	2	1	–	$r \leftarrow r \wedge A$	x		
		A, saddr	2	1	–	$A \leftarrow A \wedge (\text{saddr})$	x		
		A, !addr16	3	1	4	$A \leftarrow A \wedge (\text{addr16})$	x		
		A, [HL]	1	1	4	$A \leftarrow A \wedge (\text{HL})$	x		
		A, [HL + byte]	2	1	4	$A \leftarrow A \wedge (\text{HL} + \text{byte})$	x		
		A, [HL + B]	2	1	4	$A \leftarrow A \wedge (\text{HL} + B)$	x		
		A, [HL + C]	2	1	4	$A \leftarrow A \wedge (\text{HL} + C)$	x		
		A, ES:!addr16	4	2	5	$A \leftarrow A \wedge (\text{ES:addr16})$	x		
A, ES:[HL]		2	2	5	$A \leftarrow A \wedge (\text{ES:HL})$	x			
A, ES:[HL + byte]		3	2	5	$A \leftarrow A \wedge ((\text{ES:HL}) + \text{byte})$	x			
A, ES:[HL + B]		3	2	5	$A \leftarrow A \wedge ((\text{ES:HL}) + B)$	x			
A, ES:[HL + C]		3	2	5	$A \leftarrow A \wedge ((\text{ES:HL}) + C)$	x			

- Notes**
1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.
  2. When the program memory area is accessed.
  3. Except  $r = A$

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock ( $f_{\text{CPU}}$ ) selected by the system clock control register (CKC).
  2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).
  3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (9/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	OR	A, #byte	2	1	–	$A \leftarrow A \vee \text{byte}$		x	
		saddr, #byte	3	2	–	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$		x	
		A, r <sup>Note 3</sup>	2	1	–	$A \leftarrow A \vee r$		x	
		r, A	2	1	–	$r \leftarrow r \vee A$		x	
		A, saddr	2	1	–	$A \leftarrow A \vee (\text{saddr})$		x	
		A, !addr16	3	1	4	$A \leftarrow A \vee (\text{addr16})$		x	
		A, [HL]	1	1	4	$A \leftarrow A \vee (\text{HL})$		x	
		A, [HL + byte]	2	1	4	$A \leftarrow A \vee (\text{HL} + \text{byte})$		x	
		A, [HL + B]	2	1	4	$A \leftarrow A \vee (\text{HL} + B)$		x	
		A, [HL + C]	2	1	4	$A \leftarrow A \vee (\text{HL} + C)$		x	
		A, ES:!addr16	4	2	5	$A \leftarrow A \vee (\text{ES:addr16})$		x	
		A, ES:[HL]	2	2	5	$A \leftarrow A \vee (\text{ES:HL})$		x	
		A, ES:[HL + byte]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + \text{byte})$		x	
		A, ES:[HL + B]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + B)$		x	
	A, ES:[HL + C]	3	2	5	$A \leftarrow A \vee ((\text{ES:HL}) + C)$		x		
	XOR	A, #byte	2	1	–	$A \leftarrow A \nabla \text{byte}$		x	
		saddr, #byte	3	2	–	$(\text{saddr}) \leftarrow (\text{saddr}) \nabla \text{byte}$		x	
		A, r <sup>Note 3</sup>	2	1	–	$A \leftarrow A \nabla r$		x	
		r, A	2	1	–	$r \leftarrow r \nabla A$		x	
		A, saddr	2	1	–	$A \leftarrow A \nabla (\text{saddr})$		x	
		A, !addr16	3	1	4	$A \leftarrow A \nabla (\text{addr16})$		x	
		A, [HL]	1	1	4	$A \leftarrow A \nabla (\text{HL})$		x	
		A, [HL + byte]	2	1	4	$A \leftarrow A \nabla (\text{HL} + \text{byte})$		x	
		A, [HL + B]	2	1	4	$A \leftarrow A \nabla (\text{HL} + B)$		x	
		A, [HL + C]	2	1	4	$A \leftarrow A \nabla (\text{HL} + C)$		x	
		A, ES:!addr16	4	2	5	$A \leftarrow A \nabla (\text{ES:addr16})$		x	
		A, ES:[HL]	2	2	5	$A \leftarrow A \nabla (\text{ES:HL})$		x	
A, ES:[HL + byte]		3	2	5	$A \leftarrow A \nabla ((\text{ES:HL}) + \text{byte})$		x		
A, ES:[HL + B]	3	2	5	$A \leftarrow A \nabla ((\text{ES:HL}) + B)$		x			
A, ES:[HL + C]	3	2	5	$A \leftarrow A \nabla ((\text{ES:HL}) + C)$		x			

- Notes**
1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.
  2. When the program memory area is accessed.
  3. Except  $r = A$

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock ( $f_{\text{CPU}}$ ) selected by the system clock control register (CKC).
  2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).
  3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (10/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	–	A – byte	x	x	x
		saddr, #byte	3	1	–	(saddr) – byte	x	x	x
		A, r <sup>Note 3</sup>	2	1	–	A – r	x	x	x
		r, A	2	1	–	r – A	x	x	x
		A, saddr	2	1	–	A – (saddr)	x	x	x
		A, !addr16	3	1	4	A – (addr16)	x	x	x
		A, [HL]	1	1	4	A – (HL)	x	x	x
		A, [HL + byte]	2	1	4	A – (HL + byte)	x	x	x
		A, [HL + B]	2	1	4	A – (HL + B)	x	x	x
		A, [HL + C]	2	1	4	A – (HL + C)	x	x	x
		!addr16, #byte	4	1	4	(addr16) – byte	x	x	x
		A, ES:!addr16	4	2	5	A – (ES:addr16)	x	x	x
		A, ES:[HL]	2	2	5	A – (ES:HL)	x	x	x
		A, ES:[HL + byte]	3	2	5	A – ((ES:HL) + byte)	x	x	x
		A, ES:[HL + B]	3	2	5	A – ((ES:HL) + B)	x	x	x
		A, ES:[HL + C]	3	2	5	A – ((ES:HL) + C)	x	x	x
	ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	x	x	x	
	CMP0	A	1	1	–	A – 00H	x	x	x
		X	1	1	–	X – 00H	x	x	x
		B	1	1	–	B – 00H	x	x	x
		C	1	1	–	C – 00H	x	x	x
		saddr	2	1	–	(saddr) – 00H	x	x	x
		!addr16	3	1	4	(addr16) – 00H	x	x	x
		ES:!addr16	4	2	5	(ES:addr16) – 00H	x	x	x
	CMPS	X, [HL + byte]	3	1	4	X – (HL + byte)	x	x	x
		X, ES:[HL + byte]	4	2	5	X – ((ES:HL) + byte)	x	x	x

**Notes** 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

**Remarks** 1. One instruction clock cycle is one cycle of the CPU clock ( $f_{CPU}$ ) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).

3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (11/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	1	–	AX, CY ← AX + word	x	x	x
		AX, AX	1	1	–	AX, CY ← AX + AX	x	x	x
		AX, BC	1	1	–	AX, CY ← AX + BC	x	x	x
		AX, DE	1	1	–	AX, CY ← AX + DE	x	x	x
		AX, HL	1	1	–	AX, CY ← AX + HL	x	x	x
		AX, saddrp	2	1	–	AX, CY ← AX + (saddrp)	x	x	x
		AX, !addr16	3	1	4	AX, CY ← AX + (addr16)	x	x	x
		AX, [HL+byte]	3	1	4	AX, CY ← AX + (HL + byte)	x	x	x
		AX, ES:!addr16	4	2	5	AX, CY ← AX + (ES:addr16)	x	x	x
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX + ((ES:HL) + byte)	x	x	x
	SUBW	AX, #word	3	1	–	AX, CY ← AX – word	x	x	x
		AX, BC	1	1	–	AX, CY ← AX – BC	x	x	x
		AX, DE	1	1	–	AX, CY ← AX – DE	x	x	x
		AX, HL	1	1	–	AX, CY ← AX – HL	x	x	x
		AX, saddrp	2	1	–	AX, CY ← AX – (saddrp)	x	x	x
		AX, !addr16	3	1	4	AX, CY ← AX – (addr16)	x	x	x
		AX, [HL+byte]	3	1	4	AX, CY ← AX – (HL + byte)	x	x	x
		AX, ES:!addr16	4	2	5	AX, CY ← AX – (ES:addr16)	x	x	x
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX – ((ES:HL) + byte)	x	x	x
		CMPW	AX, #word	3	1	–	AX – word	x	x
	AX, BC		1	1	–	AX – BC	x	x	x
	AX, DE		1	1	–	AX – DE	x	x	x
	AX, HL		1	1	–	AX – HL	x	x	x
	AX, saddrp		2	1	–	AX – (saddrp)	x	x	x
	AX, !addr16		3	1	4	AX – (addr16)	x	x	x
	AX, [HL+byte]		3	1	4	AX – (HL + byte)	x	x	x
	AX, ES:!addr16		4	2	5	AX – (ES:addr16)	x	x	x
AX, ES: [HL+byte]	4		2	5	AX – ((ES:HL) + byte)	x	x	x	
Multiply	MULU	X	1	1	–	AX ← A × X			

- Notes**
1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.
  2. When the program memory area is accessed.

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock (f<sub>CPU</sub>) selected by the system clock control register (CKC).
  2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).
  3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (12/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Increment/decrement	INC	r	1	1	–	$r \leftarrow r + 1$	x	x	
		saddr	2	2	–	$(saddr) \leftarrow (saddr) + 1$	x	x	
		!addr16	3	2	–	$(addr16) \leftarrow (addr16) + 1$	x	x	
		[HL+byte]	3	2	–	$(HL+byte) \leftarrow (HL+byte) + 1$	x	x	
		ES:!addr16	4	3	–	$(ES, addr16) \leftarrow (ES, addr16) + 1$	x	x	
		ES: [HL+byte]	4	3	–	$((ES:HL) \text{ byte}) \leftarrow ((ES:HL) + \text{byte}) + 1$	x	x	
	DEC	r	1	1	–	$r \leftarrow r - 1$	x	x	
		saddr	2	2	–	$(saddr) \leftarrow (saddr) - 1$	x	x	
		!addr16	3	2	–	$(addr16) \leftarrow (addr16) - 1$	x	x	
		[HL+byte]	3	2	–	$(HL+byte) \leftarrow (HL+byte) - 1$	x	x	
		ES:!addr16	4	3	–	$(ES, addr16) \leftarrow (ES, addr16) - 1$	x	x	
		ES: [HL+byte]	4	3	–	$((ES:HL) \text{ byte}) \leftarrow ((ES:HL) + \text{byte}) - 1$	x	x	
	INCW	rp	1	1	–	$rp \leftarrow rp + 1$			
		saddrp	2	2	–	$(saddrp) \leftarrow (saddrp) + 1$			
		!addr16	3	2	–	$(addr16) \leftarrow (addr16) + 1$			
		[HL+byte]	3	2	–	$(HL+byte) \leftarrow (HL+byte) + 1$			
		ES:!addr16	4	3	–	$(ES, addr16) \leftarrow (ES, addr16) + 1$			
		ES: [HL+byte]	4	3	–	$((ES:HL) \text{ byte}) \leftarrow ((ES:HL) + \text{byte}) + 1$			
	DECW	rp	1	1	–	$rp \leftarrow rp - 1$			
		saddrp	2	2	–	$(saddrp) \leftarrow (saddrp) - 1$			
		!addr16	3	2	–	$(addr16) \leftarrow (addr16) - 1$			
		[HL+byte]	3	2	–	$(HL+byte) \leftarrow (HL+byte) - 1$			
		ES:!addr16	4	3	–	$(ES, addr16) \leftarrow (ES, addr16) - 1$			
		ES: [HL+byte]	4	3	–	$((ES:HL) \text{ byte}) \leftarrow ((ES:HL) + \text{byte}) - 1$			
Shift	SHR	A, cnt	2	1	–	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow 0) \times cnt$			x
	SHRW	AX, cnt	2	1	–	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow 0) \times cnt$			x
	SHL	A, cnt	2	1	–	$(CY \leftarrow A_7, A_m \leftarrow A_{m-1}, A_0 \leftarrow 0) \times cnt$			x
		B, cnt	2	1	–	$(CY \leftarrow B_7, B_m \leftarrow B_{m-1}, B_0 \leftarrow 0) \times cnt$			x
		C, cnt	2	1	–	$(CY \leftarrow C_7, C_m \leftarrow C_{m-1}, C_0 \leftarrow 0) \times cnt$			x
	SHLW	AX, cnt	2	1	–	$(CY \leftarrow AX_{15}, AX_m \leftarrow AX_{m-1}, AX_0 \leftarrow 0) \times cnt$			x
		BC, cnt	2	1	–	$(CY \leftarrow BC_{15}, BC_m \leftarrow BC_{m-1}, BC_0 \leftarrow 0) \times cnt$			x
	SAR	A, cnt	2	1	–	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow A_7) \times cnt$			x
SARW	AX, cnt	2	1	–	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow AX_{15}) \times cnt$			x	

**Notes 1.** When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

**2.** When the program memory area is accessed.

**Remarks 1.** One instruction clock cycle is one cycle of the CPU clock (f<sub>cpu</sub>) selected by the system clock control register (CKC).

**2.** This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).

**3.** cnt indicates the bit shift count.

**4.** In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (13/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Rotate	ROR	A, 1	2	1	–	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			×
	ROL	A, 1	2	1	–	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			×
	RORC	A, 1	2	1	–	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			×
	ROLC	A, 1	2	1	–	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			×
	ROLWC	AX,1	2	1	–	$(CY \leftarrow AX_{15}, AX_0 \leftarrow CY, AX_{m+1} \leftarrow AX_m) \times 1$			×
		BC,1	2	1	–	$(CY \leftarrow BC_{15}, BC_0 \leftarrow CY, BC_{m+1} \leftarrow BC_m) \times 1$			×
Bit manipulate	MOV1	CY, saddr.bit	3	1	–	$CY \leftarrow (\text{saddr}).\text{bit}$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow \text{sfr.bit}$			×
		CY, A.bit	2	1	–	$CY \leftarrow A.\text{bit}$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow \text{PSW.bit}$			×
		CY,[HL].bit	2	1	4	$CY \leftarrow (\text{HL}).\text{bit}$			×
		saddr.bit, CY	3	2	–	$(\text{saddr}).\text{bit} \leftarrow CY$			
		sfr.bit, CY	3	2	–	$\text{sfr.bit} \leftarrow CY$			
		A.bit, CY	2	1	–	$A.\text{bit} \leftarrow CY$			
		PSW.bit, CY	3	4	–	$\text{PSW.bit} \leftarrow CY$	×	×	
		[HL].bit, CY	2	2	–	$(\text{HL}).\text{bit} \leftarrow CY$			
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow (\text{ES}, \text{HL}).\text{bit}$			×
	ES:[HL].bit, CY	3	3	–	$(\text{ES}, \text{HL}).\text{bit} \leftarrow CY$				
	AND1	CY, saddr.bit	3	1	–	$CY \leftarrow CY \wedge (\text{saddr}).\text{bit}$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow CY \wedge \text{sfr.bit}$			×
		CY, A.bit	2	1	–	$CY \leftarrow CY \wedge A.\text{bit}$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \wedge \text{PSW.bit}$			×
		CY,[HL].bit	2	1	4	$CY \leftarrow CY \wedge (\text{HL}).\text{bit}$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \wedge (\text{ES}, \text{HL}).\text{bit}$			×
	OR1	CY, saddr.bit	3	1	–	$CY \leftarrow CY \vee (\text{saddr}).\text{bit}$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow CY \vee \text{sfr.bit}$			×
		CY, A.bit	2	1	–	$CY \leftarrow CY \vee A.\text{bit}$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \vee \text{PSW.bit}$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \vee (\text{HL}).\text{bit}$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \vee (\text{ES}, \text{HL}).\text{bit}$			×

- Notes**
1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.
  2. When the program memory area is accessed.

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock ( $f_{\text{CPU}}$ ) selected by the system clock control register (CKC).
  2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).
  3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (14/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, saddr.bit	3	1	–	$CY \leftarrow CY \nabla (\text{saddr}).\text{bit}$			×
		CY, sfr.bit	3	1	–	$CY \leftarrow CY \nabla \text{sfr}.\text{bit}$			×
		CY, A.bit	2	1	–	$CY \leftarrow CY \nabla A.\text{bit}$			×
		CY, PSW.bit	3	1	–	$CY \leftarrow CY \nabla \text{PSW}.\text{bit}$			×
		CY, [HL].bit	2	1	4	$CY \leftarrow CY \nabla (\text{HL}).\text{bit}$			×
		CY, ES:[HL].bit	3	2	5	$CY \leftarrow CY \nabla (\text{ES}, \text{HL}).\text{bit}$			×
	SET1	saddr.bit	3	2	–	$(\text{saddr}).\text{bit} \leftarrow 1$			
		sfr.bit	3	2	–	$\text{sfr}.\text{bit} \leftarrow 1$			
		A.bit	2	1	–	$A.\text{bit} \leftarrow 1$			
		!addr16.bit	4	2	–	$(\text{addr}16).\text{bit} \leftarrow 1$			
		PSW.bit	3	4	–	$\text{PSW}.\text{bit} \leftarrow 1$	×	×	×
		[HL].bit	2	2	–	$(\text{HL}).\text{bit} \leftarrow 1$			
		ES:!addr16.bit	5	3	–	$(\text{ES}, \text{addr}16).\text{bit} \leftarrow 1$			
		ES:[HL].bit	3	3	–	$(\text{ES}, \text{HL}).\text{bit} \leftarrow 1$			
	CLR1	saddr.bit	3	2	–	$(\text{saddr}.\text{bit}) \leftarrow 0$			
		sfr.bit	3	2	–	$\text{sfr}.\text{bit} \leftarrow 0$			
		A.bit	2	1	–	$A.\text{bit} \leftarrow 0$			
		!addr16.bit	4	2	–	$(\text{addr}16).\text{bit} \leftarrow 0$			
		PSW.bit	3	4	–	$\text{PSW}.\text{bit} \leftarrow 0$	×	×	×
		[HL].bit	2	2	–	$(\text{HL}).\text{bit} \leftarrow 0$			
		ES:!addr16.bit	5	3	–	$(\text{ES}, \text{addr}16).\text{bit} \leftarrow 0$			
		ES:[HL].bit	3	3	–	$(\text{ES}, \text{HL}).\text{bit} \leftarrow 0$			
	SET1	CY	2	1	–	$CY \leftarrow 1$			1
	CLR1	CY	2	1	–	$CY \leftarrow 0$			0
	NOT1	CY	2	1	–	$CY \leftarrow \overline{CY}$			×

**Notes 1.** When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

**2.** When the program memory area is accessed.

**Remarks 1.** One instruction clock cycle is one cycle of the CPU clock ( $f_{\text{CPU}}$ ) selected by the system clock control register (CKC).

**2.** This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).

**3.** In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (15/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/ return	CALL	rp	2	3	–	$(SP - 2) \leftarrow (PC + 2)_S$ , $(SP - 3) \leftarrow (PC + 2)_H$ , $(SP - 4) \leftarrow (PC + 2)_L$ , $PC \leftarrow CS$ , rp, $SP \leftarrow SP - 4$			
		\$!addr20	3	3	–	$(SP - 2) \leftarrow (PC + 3)_S$ , $(SP - 3) \leftarrow (PC + 3)_H$ , $(SP - 4) \leftarrow (PC + 3)_L$ , $PC \leftarrow PC + 3 +$ jdisp16, $SP \leftarrow SP - 4$			
		!addr16	3	3	–	$(SP - 2) \leftarrow (PC + 3)_S$ , $(SP - 3) \leftarrow (PC + 3)_H$ , $(SP - 4) \leftarrow (PC + 3)_L$ , $PC \leftarrow 0000$ , addr16, $SP \leftarrow SP - 4$			
		!!addr20	4	3	–	$(SP - 2) \leftarrow (PC + 4)_S$ , $(SP - 3) \leftarrow (PC + 4)_H$ , $(SP - 4) \leftarrow (PC + 4)_L$ , $PC \leftarrow addr20$ , $SP \leftarrow SP - 4$			
	CALLT	[addr5]	2	5	–	$(SP - 2) \leftarrow (PC + 2)_S$ , $(SP - 3) \leftarrow (PC + 2)_H$ , $(SP - 4) \leftarrow (PC + 2)_L$ , $PC_S \leftarrow 0000$ , $PC_H \leftarrow (0000, addr5 + 1)$ , $PC_L \leftarrow (0000, addr5)$ , $SP \leftarrow SP - 4$			
	BRK	–	2	5	–	$(SP - 1) \leftarrow PSW$ , $(SP - 2) \leftarrow (PC + 2)_S$ , $(SP - 3) \leftarrow (PC + 2)_H$ , $(SP - 4) \leftarrow (PC + 2)_L$ , $PC_S \leftarrow 0000$ , $PC_H \leftarrow (0007FH)$ , $PC_L \leftarrow (0007EH)$ , $SP \leftarrow SP - 4$ , $IE \leftarrow 0$			
	RET	–	1	6	–	$PC_L \leftarrow (SP)$ , $PC_H \leftarrow (SP + 1)$ , $PC_S \leftarrow (SP + 2)$ , $SP \leftarrow SP + 4$			
RETI	–	2	6	–	$PC_L \leftarrow (SP)$ , $PC_H \leftarrow (SP + 1)$ , $PC_S \leftarrow (SP + 2)$ , $PSW \leftarrow (SP + 3)$ , $SP \leftarrow SP + 4$	R	R	R	
RETB	–	2	6	–	$PC_L \leftarrow (SP)$ , $PC_H \leftarrow (SP + 1)$ , $PC_S \leftarrow (SP + 2)$ , $PSW \leftarrow (SP + 3)$ , $SP \leftarrow SP + 4$	R	R	R	

**Notes** 1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

**Remarks** 1. One instruction clock cycle is one cycle of the CPU clock ( $f_{CPU}$ ) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).

3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.



Table 28-5. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulate	PUSH	PSW	2	1	–	(SP – 1) ← PSW, (SP – 2) ← 00H, SP ← SP – 2			
		rp	1	1	–	(SP – 1) ← rpH, (SP – 2) ← rpL, SP ← SP – 2			
	POP	PSW	2	3	–	PSW ← (SP + 1), SP ← SP + 2	R	R	R
		rp	1	1	–	rpL ← (SP), rpH ← (SP + 1), SP ← SP + 2			
	MOVW	SP, #word	4	1	–	SP ← word			
		SP, AX	2	1	–	SP ← AX			
		AX, SP	2	1	–	AX ← SP			
		HL, SP	3	1	–	HL ← SP			
		BC, SP	3	1	–	BC ← SP			
		DE, SP	3	1	–	DE ← SP			
	ADDW	SP, #byte	2	1	–	SP ← SP + byte			
SUBW	SP, #byte	2	1	–	SP ← SP – byte				
Unconditional branch	BR	AX	2	3	–	PC ← CS, AX			
		\$addr20	2	3	–	PC ← PC + 2 + jdisp8			
		!\$addr20	3	3	–	PC ← PC + 3 + jdisp16			
		!addr16	3	3	–	PC ← 0000, addr16			
		!!addr20	4	3	–	PC ← addr20			
Conditional branch	BC	\$addr20	2	2/4 <sup>Note 3</sup>	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	\$addr20	2	2/4 <sup>Note 3</sup>	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	\$addr20	2	2/4 <sup>Note 3</sup>	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	\$addr20	2	2/4 <sup>Note 3</sup>	–	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	\$addr20	3	2/4 <sup>Note 3</sup>	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=0			
	BNH	\$addr20	3	2/4 <sup>Note 3</sup>	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=1			
	BT	saddr.bit, \$addr20	4	3/5 <sup>Note 3</sup>	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, \$addr20	4	3/5 <sup>Note 3</sup>	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, \$addr20	3	3/5 <sup>Note 3</sup>	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, \$addr20	4	3/5 <sup>Note 3</sup>	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
[HL].bit, \$addr20		3	3/5 <sup>Note 3</sup>	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1				
	ES:[HL].bit, \$addr20	4	4/6 <sup>Note 3</sup>	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1				

<R>  
<R>

- Notes**
1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.
  2. When the program memory area is accessed.
  3. This indicates the number of clocks “when condition is not met/when condition is met”.

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock (f<sub>cpu</sub>) selected by the system clock control register (CKC).
  2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).
  3. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

Table 28-5. Operation List (17/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
<R> <R>	Conditional branch	saddr.bit, \$addr20	4	3/5 <sup>Note 3</sup>	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, \$addr20	4	3/5 <sup>Note 3</sup>	–	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, \$addr20	3	3/5 <sup>Note 3</sup>	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, \$addr20	4	3/5 <sup>Note 3</sup>	–	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, \$addr20	3	3/5 <sup>Note 3</sup>	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, \$addr20	4	4/6 <sup>Note 3</sup>	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, \$addr20	4	3/5 <sup>Note 3</sup>	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, \$addr20	4	3/5 <sup>Note 3</sup>	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, \$addr20	3	3/5 <sup>Note 3</sup>	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, \$addr20	4	3/5 <sup>Note 3</sup>	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, \$addr20	3	3/5 <sup>Note 3</sup>	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, \$addr20	4	4/6 <sup>Note 3</sup>	–	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit			
Conditional skip	SKC	–	2	1	–	Next instruction skip if CY = 1			
	SKNC	–	2	1	–	Next instruction skip if CY = 0			
	SKZ	–	2	1	–	Next instruction skip if Z = 1			
	SKNZ	–	2	1	–	Next instruction skip if Z = 0			
	SKH	–	2	1	–	Next instruction skip if (Z ∨ CY) = 0			
	SKNH	–	2	1	–	Next instruction skip if (Z ∨ CY) = 1			
CPU control	SEL	Rb <sub>n</sub>	2	1	–	RBS[1:0] ← n			
	NOP	–	1	1	–	No Operation			
	EI	–	3	4	–	IE ← 1(Enable Interrupt)			
	DI	–	3	4	–	IE ← 0(Disable Interrupt)			
	HALT	–	2	3	–	Set HALT Mode			
	STOP	–	2	3	–	Set STOP Mode			

- Notes**
1. When the internal RAM area, SFR area, or extended SFR area is accessed, or for an instruction with no data access.
  2. When the program memory area is accessed.
  3. This indicates the number of clocks “when condition is not met/when condition is met”.

- Remarks**
1. One instruction clock cycle is one cycle of the CPU clock ( $f_{CPU}$ ) selected by the system clock control register (CKC).
  2. This number of clocks is for when the program is in the internal ROM (flash memory) area. When fetching an instruction from the internal RAM area, the number of clocks is twice the number of clocks plus 3, maximum (except when branching to the external memory area).
  3. n indicates the number of register banks (n = 0 to 3)
  4. In products where the external memory area is adjacent to the internal flash area, the number of waits is added to the number of instruction execution clocks placed in the last address (16-byte max.) in the flash memory, in order to use the external bus interface function. This should be done because, during pre-reading of the instruction code, an external memory wait being inserted due to an external memory area exceeding the flash space is accessed. For the number of waits, refer to **5.4 Number of Instruction Wait Clocks for Data Access**.

## CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)

Target products Conventional-specification products:  $\mu$ PD78F1174, 78F1175, 78F1176, 78F1177, 78F1178  
 Expanded-specification products:  $\mu$ PD78F1174A, 78F1175A, 78F1176A, 78F1177A, 78F1178A

**Caution** The 78K0R/KH3 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. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.

### Absolute Maximum Ratings (T<sub>A</sub> = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V <sub>DD</sub>		-0.5 to +6.5	V
	EV <sub>DD0</sub> , EV <sub>DD1</sub>	EV <sub>DD0</sub> = EV <sub>DD1</sub>	-0.5 to +6.5	V
	V <sub>SS</sub>		-0.5 to +0.3	V
	EV <sub>SS0</sub> , EV <sub>SS1</sub>	EV <sub>SS0</sub> = EV <sub>SS1</sub>	-0.5 to +0.3	V
	AV <sub>REF0</sub>		-0.5 to V <sub>DD</sub> +0.3 <sup>Note 1</sup>	V
	AV <sub>REF1</sub>		-0.5 to V <sub>DD</sub> +0.3 <sup>Note 1</sup>	V
	AV <sub>SS</sub>		-0.5 to +0.3	V
REGC pin input voltage	V <sub>IREGC</sub>	REGC	-0.3 to +3.6 and -0.3 to V <sub>DD</sub> +0.3 <sup>Note 2</sup>	V
Input voltage	V <sub>I1</sub>	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120 to P127, P131, P140 to P145, P160 to P163, EXCLK, RESET, FLMD0	-0.3 to EV <sub>DD0</sub> , EV <sub>DD1</sub> +0.3 and -0.3 to V <sub>DD</sub> +0.3 <sup>Note 1</sup>	V
	V <sub>I2</sub>	P60 to P63 (N-ch open-drain)	-0.3 to +6.5	V
	V <sub>I3</sub>	P20 to P27, P150 to P157	-0.3 to AV <sub>REF0</sub> +0.3 and -0.3 to V <sub>DD</sub> +0.3 <sup>Note 1</sup>	V
	V <sub>I4</sub>	P110, P111	-0.3 to AV <sub>REF1</sub> +0.3 and -0.3 to V <sub>DD</sub> +0.3 <sup>Note 1</sup>	V
Output voltage	V <sub>O1</sub>	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	-0.3 to EV <sub>DD0</sub> , EV <sub>DD1</sub> +0.3 <sup>Note 1</sup>	V
	V <sub>O2</sub>	P20 to P27, P150 to P157	-0.3 to AV <sub>REF0</sub> +0.3	V
	V <sub>O3</sub>	P110, P111	-0.3 to AV <sub>REF1</sub> +0.3	V

**Notes 1.** Must be 6.5 V or lower.

- 2.** Connect the REGC pin to V<sub>SS</sub> via a capacitor (0.47 to 1  $\mu$ F). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

**Caution** Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**Absolute Maximum Ratings (T<sub>A</sub> = 25°C) (2/2)**

Parameter	Symbols	Conditions		Ratings	Unit
Analog input voltage	V <sub>AN</sub>	ANI0 to ANI15		-0.3 to AV <sub>REF0</sub> +0.3 <sup>Note</sup> and -0.3 to V <sub>DD</sub> +0.3 <sup>Note</sup>	V
Analog output voltage	V <sub>AO</sub>	ANO0, ANO1		-0.3 to AV <sub>REF1</sub> +0.3	V
Output current, high	I <sub>OH1</sub>	Per pin	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	-10	mA
		Total of all pins -80 mA	P00 to P04, P07, P32 to P37, P40 to P47, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	-25	mA
			P05, P06, P10 to P17, P30, P31, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117	-55	mA
	I <sub>OH2</sub>	Per pin	P20 to P27, P110, P111, P150 to P157	-0.5	mA
		Total of all pins		-2	mA
Output current, low	I <sub>OL1</sub>	Per pin	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	30	mA
		Total of all pins 200 mA	P00 to P04, P07, P32 to P37, P40 to P47, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	60	mA
			P05, P06, P10 to P17, P30, P31, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117	140	mA
	I <sub>OL2</sub>	Per pin	P20 to P27, P110, P111, P150 to P157	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T <sub>A</sub>	In normal operation mode		-40 to +85	°C
		In flash memory programming mode			
Storage temperature	T <sub>stg</sub>			-65 to +150	°C

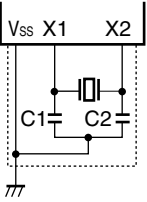
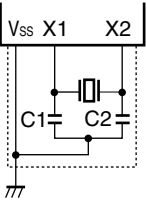
**Note** Must be 6.5 V or lower.

**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.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

&lt;R&gt;

**X1 Oscillator Characteristics****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )**

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency ( $f_x$ ) <sup>Note</sup>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		20.0	MHz
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	2.0		5.0	
Crystal resonator		X1 clock oscillation frequency ( $f_x$ ) <sup>Note</sup>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		20.0	MHz
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	2.0		5.0	

**Note** Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

**Cautions 1.** When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- Keep the wiring length as short as possible.
  - Do not cross the wiring with the other signal lines.
  - Do not route the wiring near a signal line through which a high fluctuating current flows.
  - Always make the ground point of the oscillator capacitor the same potential as  $V_{SS}$ .
  - Do not ground the capacitor to a ground pattern through which a high current flows.
  - Do not fetch signals from the oscillator.
2. Since the CPU is started by the internal high-speed oscillation clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

**Internal Oscillator Characteristics****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )**

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit	
8 MHz internal oscillator	Internal high-speed oscillation clock frequency ( $f_{IH}$ ) <sup>Note 1</sup>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	7.6	8.0	8.4	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	5.0	8.0	8.4	MHz	
240 kHz internal oscillator	Internal low-speed oscillation clock frequency ( $f_{IL}$ )	Normal current mode	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	216	240	264	kHz
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	192	240	264	kHz
		Low consumption current mode <sup>Note 2</sup>	192	240	264	kHz	

**Notes 1.** This only indicates the oscillator characteristics of when HIOTRM is set to 10H. Refer to AC Characteristics for instruction execution time.

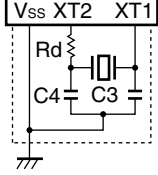
**2.** Regulator output is set to low consumption current mode in the following cases:

- When the RMC register is set to 5AH.
- During system reset.
- In STOP mode (except during OCD mode).
- When both the high-speed system clock ( $f_{MX}$ ) and the high-speed internal oscillation clock ( $f_{IH}$ ) are stopped during CPU operation with the subsystem clock ( $f_{XT}$ ).
- When both the high-speed system clock ( $f_{MX}$ ) and the high-speed internal oscillation clock ( $f_{IH}$ ) are stopped during the HALT mode when the CPU operation with the subsystem clock ( $f_{XT}$ ) has been set.

**Remark** For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 23 REGULATOR**.

&lt;R&gt;

**XT1 Oscillator Characteristics****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )**

Resonator	Recommended Circuit	Items	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency ( $f_{XT1}$ ) <sup>Note</sup>		32	32.768	35	kHz

**Note** Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

**Cautions 1.** When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.

## Recommended Oscillator Constants

(1) X1 oscillation: Ceramic resonator (AMPH = 0, RMC = 00H, T<sub>A</sub> = -40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants		Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	MIN. (V)	MAX. (V)
Murata Manufacturing Co., Ltd.	CSTCC2M00G56-R0	SMD	2.0	Internal (47)	Internal (47)	1.8	5.5
	CSTCR4M00G55-R0	SMD	4.0	Internal (39)	Internal (39)	1.8	
	CSTLS4M00G56-B0	Lead		Internal (47)	Internal (47)	1.8	
	CSTCR4M19G55-R0	SMD	4.194	Internal (39)	Internal (39)	1.8	
	CSTLS4M19G56-B0	Lead		Internal (47)	Internal (47)	1.8	
	CSTCR4M91G55-R0	SMD	4.915	Internal (39)	Internal (39)	1.8	
	CSTLS4M91G53-B0	Lead		Internal (15)	Internal (15)	1.8	
	CSTLS4M91G56-B0			Internal (47)	Internal (47)	2.1	
	CSTCR5M00G53-R0	SMD	5.0	Internal (15)	Internal (15)	1.8	
	CSTCR5M00G55-R0			Internal (39)	Internal (39)	1.8	
	CSTLS5M00G53-B0	Lead	5.0	Internal (15)	Internal (15)	1.8	
	CSTLS5M00G56-B0			Internal (47)	Internal (47)	2.1	
	CSTCR6M00G53-R0	SMD	6.0	Internal (15)	Internal (15)	1.8	
	CSTCR6M00G55-R0			Internal (39)	Internal (39)	1.9	
	CSTLS6M00G53-B0	Lead	6.0	Internal (15)	Internal (15)	1.8	
	CSTLS6M00G56-B0			Internal (47)	Internal (47)	2.2	
	CSTCE8M00G52-R0	SMD	8.0	Internal (10)	Internal (10)	1.8	
	CSTCE8M00G55-R0			Internal (33)	Internal (33)	1.9	
	CSTLS8M00G53-B0	Lead	8.0	Internal (15)	Internal (15)	1.8	
	CSTLS8M00G56-B0			Internal (47)	Internal (47)	2.4	
	CSTCE8M38G52-R0	SMD	8.388	Internal (10)	Internal (10)	1.8	
	CSTCE8M38G55-R0			Internal (33)	Internal (33)	1.9	
	CSTLS8M38G53-B0	Lead	8.388	Internal (15)	Internal (15)	1.8	
	CSTLS8M38G56-B0			Internal (47)	Internal (47)	2.4	
CSTCE10M0G52-R0	SMD	10.0	Internal (10)	Internal (10)	1.8		
CSTCE10M0G55-R0			Internal (33)	Internal (33)	2.1		
CSTLS10M0G53-B0	Lead	10.0	Internal (15)	Internal (15)	1.8		
TOKO, Inc.	DCRHTC(P)2.00LL	Lead	2.0	Internal (30)	Internal (30)	1.8	5.5
	DCRHTC(P)4.00LL		4.0	Internal (30)	Internal (30)		
	DECRHTC4.00	SMD	4.0	Internal (15)	Internal (15)		
	DCRHYC(P)8.00A	Lead	8.0	Internal (22)	Internal (22)		

**Caution** The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

<R> When doing so, check the conditions for using the AMPH bit, RMC register, and whether to enter or exit the STOP mode.

The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.



<R> (2) X1 oscillation: Crystal resonator (AMPH = 0, RMC = 00H, T<sub>A</sub> = -40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants		Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	MIN. (V)	MAX. (V)
KYOCERA KINSEKI Co., Ltd.	HC49SFWB04194D0PPTZZ	Lead	4.194	10	10	1.8	5.5
	CX49GFWB04194D0PPTZZ						
	CX1255GB04194D0PPTZZ	SMD					
	HC49SFWB05000D0PPTZZ	Lead	5.0	10	10	1.8	
	CX49GFWB05000D0PPTZZ						
	CX1255GB05000D0PPTZZ	SMD					
	CX8045GB05000D0PPTZZ						
	HC49SFWB08380D0PPTZZ	Lead	8.38	10	10	1.8	
	CX49GFWB08380D0PPTZZ						
	CX1255GB08380D0PPTZZ	SMD					
	CX8045GB08380D0PPTZZ						
	CX5032GB08380D0PPTZZ						
	HC49SFWB10000D0PPTZZ	Lead	10.0	10	10	1.8	
	CX49GFWB10000D0PPTZZ						
	CX1255GB10000D0PPTZZ	SMD					
	CX8045GB10000D0PPTZZ						
	CX5032GB10000D0PPTZZ						
	CX5032SB10000D0PPTZZ						
CX3225GB10000D0PPTZZ							

**Caution** The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

When doing so, check the conditions for using the AMPH bit, RMC register, and whether to enter or exit the STOP mode.

The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.

**(3) X1 oscillation: Ceramic resonator (AMPH = 1, RMC = 00H, T<sub>A</sub> = -40 to +85°C)**

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants		Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	MIN. (V)	MAX. (V)
Murata Manufacturing Co., Ltd.	CSTCE12M0G55-R0	SMD	12.0	Internal (33)	Internal (33)	1.8	5.5
	CSTCE16M0V53-R0	SMD	16.0	Internal (15)	Internal (15)	1.8	
	CSTLS16M0X51-B0	Lead		Internal (5)	Internal (5)	1.8	
	CSTCE20M0V53-R0	SMD	20.0	Internal (15)	Internal (15)	1.9	
	CSTCG20M0V53-R0	Small SMD		Internal (15)	Internal (15)	2.0	
	CSTLS20M0X51-B0	Lead		Internal (5)	Internal (5)	1.9	
TOKO, Inc.	DCRHYC(P)12.00A	Lead		12.0	Internal (22)	Internal (22)	1.8
	DCRHZ(P)16.00A-15	Lead	16.0	Internal (15)	Internal (15)		
	DCRHZ(P)20.00A-15	Lead	20.0	Internal (15)	Internal (15)	2.0	
	DECRHZ20.00	SMD		Internal (10)	Internal (10)	1.8	

**Caution** The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

<R> When doing so, check the conditions for using the AMPH bit, RMC register, and whether to enter or exit the STOP mode.

The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.

&lt;R&gt;

**(4) X1 oscillation: Crystal resonator (AMPH = 1, RMC = 00H, T<sub>A</sub> = -40 to +85°C)**

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants		Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	MIN. (V)	MAX. (V)
KYOCERA KINSEKI Co., Ltd.	HC49SFWB16000D0PPTZZ	Lead	16.0	10	10	1.8	5.5
	CX49GFWB16000D0PPTZZ						
	CX1255GB16000D0PPTZZ	SMD					
	CX8045GB16000D0PPTZZ						
	CX5032GB16000D0PPTZZ						
	CX5032SB16000D0PPTZZ						
	CX3225GB16000D0PPTZZ						
	CX3225SB16000D0PPTZZ						
	CX2520SB16000D0PPTZZ						
	HC49SFWB20000D0PPTZZ						
	CX49GFWB20000D0PPTZZ						
	CX1255GB20000D0PPTZZ	SMD					
	CX8045GB20000D0PPTZZ						
	CX5032GB20000D0PPTZZ						
	CX5032SB20000D0PPTZZ						
	CX3225GB20000D0PPTZZ						
	CX3225SB20000D0PPTZZ						
	CX2520SB20000D0PPTZZ						
	CX2016SB20000D0PPTZZ						

**Caution** The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

When doing so, check the conditions for using the AMPH bit, RMC register, and whether to enter or exit the STOP mode.

The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.

**(5) XT1 oscillation: Crystal resonator ( $T_A = -40$  to  $+85^\circ\text{C}$ )**

Manufacturer	Part Number	SMD/Lead	Frequency (kHz)	Load Capacitance CL (pF)	Recommended Circuit Constants			Oscillation Voltage Range	
					C3 (pF)	C4 (pF)	Rd (k $\Omega$ )	MIN. (V)	MAX. (V)
Seiko Instruments Inc.	SP-T2A	SMD	32.768	6.0	5	5	0	1.8	5.5
				12.5	18	18	0		
	SSP-T7	Small SMD		7.0	7	7	0		
				12.5	18	18	0		
	VT-200	Lead		6.0	5	5	0		
				12.5	18	18	0		
CITIZEN FINETECH MIYOTA CO., LTD.	CM200S	SMD	32.768	9.0	12	15	0	1.8	5.5
					12	15	100		
	CM315	SMD		9.0	15	15	0		
					15	15	100		
	CM519	SMD		9.0	15	12	0		
					15	12	100		

**Caution** The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

<R> When doing so, check the conditions for using the RMC register, and whether to enter or exit the STOP mode.

The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.

**(6) XT1 oscillation: Crystal resonator ( $T_A = -20$  to  $+70^\circ\text{C}$ )**

Manufacturer	Part Number	SMD/Lead	Frequency (kHz)	Load Capacitance CL (pF)	Recommended Circuit Constants			Oscillation Voltage Range	
					C3 (pF)	C4 (pF)	Rd (k $\Omega$ )	MIN. (V)	MAX. (V)
CITIZEN FINETECH MIYOTA CO., LTD.	CFS-206	Lead	32.768	12.5	22	18	0	1.8	5.5
					22	18	100		
				9.0	12	15	0		
					12	15	100		

**Caution** The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

<R> When doing so, check the conditions for using the RMC register, and whether to enter or exit the STOP mode.

The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.

**DC Characteristics (1/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Output current, high <sup>Note 1</sup>	$I_{OH1}$	Per pin for P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-3.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-1.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-1.0	mA
		Total of P00 to P04, P07, P32 to P37, P40 to P47, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163 (When duty = 70% <sup>Note 2</sup> )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-20.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-10.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-5.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117 (When duty = 70% <sup>Note 2</sup> )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-30.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-19.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-10.0	mA
	Total of all pins (When duty = 60% <sup>Note 2</sup> )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-50.0	mA	
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-29.0	mA	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-15.0	mA	
$I_{OH2}$	Per pin for P20 to P27, P150 to P157	$AV_{REF0} \leq V_{DD}$			-0.1	mA	
	Per pin for P110, P111	$AV_{REF1} \leq V_{DD}$			-0.1	mA	

**Notes** 1. Value of current at which the device operation is guaranteed even if the current flows from  $EV_{DD0}$  or  $EV_{DD1}$  pin to an output pin.

2. Specification under conditions where the duty factor is 60% or 70%.

The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins =  $(I_{OH} \times 0.7)/(n \times 0.01)$

<Example> Where  $n = 50\%$  and  $I_{OH} = -20.0\text{ mA}$

$$\text{Total output current of pins} = (-20.0 \times 0.7)/(50 \times 0.01) = -28.0\text{ mA}$$

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

**Caution** P02 to P04, P43, P45, P95 to P97, P125 to P127, and P142 to P144 do not output high level in N-ch open-drain mode.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**DC Characteristics (2/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  
 $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output current, low <sup>Note 1</sup>	IoL1	Per pin for P00 to P02, P05 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P94, P97, P112 to P117, P120, P127, P130, P131, P140, P141, P144, P145, P160 to P163	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		8.5	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		1.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		0.5	mA
		Per pin for P03, P04, P95, P96, P125, P126, P142, P143	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		8.5	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		1.5	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		0.6	mA
		Per pin for P60 to P63	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		15.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		3.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		2.0	mA
		Total of P00 to P04, P07, P32 to P37, P40 to P47, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163 (When duty = 70% <sup>Note 2</sup> )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		20.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		15.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		9.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117 (When duty = 70% <sup>Note 2</sup> )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		45.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		35.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		20.0	mA
Total of all pins (When duty = 60% <sup>Note 2</sup> )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		65.0	mA		
	$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$		50.0	mA		
	$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		29.0	mA		
IoL2	Per pin for P20 to P27, P150 to P157	$AV_{REF0} \leq V_{DD}$			0.4	mA
	P110, P111	$AV_{REF1} \leq V_{DD}$			0.4	mA

**Notes** 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to  $EV_{SS0}$ ,  $EV_{SS1}$ ,  $V_{SS}$ , or  $AV_{SS}$  pin.

2. Specification under conditions where the duty factor is 60% or 70%.

The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins =  $(IoL \times 0.7)/(n \times 0.01)$

<Example> Where  $n = 50\%$  and  $IoL = 20.0\text{ mA}$

$$\text{Total output current of pins} = (20.0 \times 0.7)/(50 \times 0.01) = 28.0\text{ mA}$$

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**DC Characteristics (3/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input voltage, high	$V_{IH1}$	P01, P02, P12, P13, P15, P32 to P37, P41, P45, P52 to P57, P64 to P67, P80 to P87, P90 to P94, P97, P112 to P117, P121 to P124, P127, P144	$0.7V_{DD}$		$V_{DD}$	V	
	$V_{IH2}$	P00, P03 to P06, P10, P11, P14, P16, P17, P30, P31, P40, P42 to P44, P46, P47, P50, P51, P70 to P77, P95, P96, P120, P125, P126, P131, P140 to P143, P145, P160 to P163, EXCLK, RESET	Normal input buffer $0.8V_{DD}$		$V_{DD}$	V	
	$V_{IH3}$	P03, P04, P43, P44, P95, P96, P125, P126, P142, P143	TTL input buffer $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.2		$V_{DD}$	V
			TTL input buffer $2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	2.0		$V_{DD}$	V
			TTL input buffer $1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	1.6		$V_{DD}$	V
	$V_{IH4}$	P20 to P27, P150 to P157	$2.7\text{ V} \leq AV_{REF0} \leq V_{DD}$	$0.7AV_{REF0}$		$AV_{REF0}$	V
			$AV_{REF0} = V_{DD} < 2.7\text{ V}$				
	$V_{IH5}$	P110, P111	$2.7\text{ V} \leq AV_{REF1} \leq V_{DD}$	$0.7AV_{REF1}$		$AV_{REF1}$	V
			$AV_{REF1} = V_{DD} < 2.7\text{ V}$				
	$V_{IH6}$	P60 to P63		$0.7V_{DD}$		6.0	V
$V_{IH7}$	FLMD0		$0.9V_{DD}$ Note		$V_{DD}$	V	

**Note** Must be  $0.9V_{DD}$  or higher when used in the flash memory programming mode.

- Cautions**
- The maximum value of  $V_{IH}$  of pins P02 to P04, P43, P45, P95 to P97, P125 to P127, and P142 to P144 is  $V_{DD}$ , even in the N-ch open-drain mode.
  - For P122/EXCLK, the value of  $V_{IH}$  differ according to the input port mode or external clock mode. Make sure to satisfy the DC characteristics of EXCLK in external clock input mode.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**DC Characteristics (4/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input voltage, low	$V_{IL1}$	P01, P02, P07, P12, P13, P15, P32 to P37, P41, P45, P52 to P57, P64 to P67, P80 to P87, P90 to P94, P97, P112 to P117, P121 to P124, P127, P144	0		$0.3V_{DD}$	V	
	$V_{IL2}$	P00, P03 to P06, P10, P11, P14, P16, P17, P30, P31, P40, P42 to P44, P46, P47, P50, P51, P70 to P77, P95, P96, P120, P125, P126, P131, P140 to P143, P145, P160 to P163, EXCLK, RESET	0		$0.2V_{DD}$	V	
	$V_{IL3}$	P03, P04, P43, P44, P95, P96, P125, P126, P142, P143	TTL input buffer $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0		0.8	V
			TTL input buffer $2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	0		0.5	V
			TTL input buffer $1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	0		0.2	V
	$V_{IL4}$	P20 to P27, P150 to P157	$2.7\text{ V} \leq AV_{REF0} \leq V_{DD}$	0		$0.3AV_{REF0}$	V
			$AV_{REF0} = V_{DD} < 2.7\text{ V}$				
	$V_{IL5}$	P110, P111	$2.7\text{ V} \leq AV_{REF1} \leq V_{DD}$	0		$0.3AV_{REF1}$	V
			$AV_{REF1} = V_{DD} < 2.7\text{ V}$				
	$V_{IL6}$	P60 to P63		0		$0.3V_{DD}$	V
$V_{IL7}$	FLMD0 <sup>Note</sup>		0		$0.1V_{DD}$	V	

**Note** When disabling writing of the flash memory, connect the FLMD0 pin processing directly to  $V_{SS}$ , and maintain a voltage less than  $0.1V_{DD}$ .

**Caution** For P122/EXCLK, the value of  $V_{IL}$  differ according to the input port mode or external clock mode. Make sure to satisfy the DC characteristics of EXCLK in external clock input mode.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.



**DC Characteristics (5/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Output voltage, high	$V_{OH1}$	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OH1} = -3.0\text{ mA}$	$V_{DD} - 0.7$			V
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OH1} = -1.0\text{ mA}$	$V_{DD} - 0.5$			V
	$V_{OH2}$	P20 to P27, P150 to P157	$AV_{REF0} \leq V_{DD}$ , $I_{OH2} = -0.1\text{ mA}$	$AV_{REF0} - 0.5$			V
		P110, P111	$AV_{REF1} \leq V_{DD}$ , $I_{OH2} = -0.1\text{ mA}$	$AV_{REF1} - 0.5$			V
Output voltage, low	$V_{OL1}$	P00 to P02, P05 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P94, P97, P112 to P117, P120, P127, P130, P131, P140, P141, P144, P145, P160 to P163	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 8.5\text{ mA}$			0.7	V
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 1.0\text{ mA}$			0.5	V
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 0.5\text{ mA}$			0.4	V
		P03, P04, P95, P96, P125, P126, P142, P143	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 8.5\text{ mA}$			0.7	V
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 1.5\text{ mA}$			0.5	V
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 0.6\text{ mA}$			0.4	V
		$V_{OL2}$	P20 to P27, P150 to P157	$AV_{REF0} \leq V_{DD}$ , $I_{OL2} = 0.4\text{ mA}$			0.4
	$AV_{REF1} \leq V_{DD}$ , $I_{OL2} = 0.4\text{ mA}$					0.4	V
	$V_{OL3}$	P60 to P63	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 15.0\text{ mA}$			2.0	V
			$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 5.0\text{ mA}$			0.4	V
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 3.0\text{ mA}$			0.4	V
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 2.0\text{ mA}$			0.4	V

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**DC Characteristics (6/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input leakage current, high	I <sub>LIH1</sub>	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P131, P140 to P145, P160 to P163, FLMD0, $\overline{\text{RESET}}$	$V_i = V_{DD}$			1	$\mu\text{A}$
	I <sub>LIH2</sub>	P20 to P27, P150 to P157	$V_i = AV_{REF0}$ , $2.7\text{ V} \leq AV_{REF0} \leq V_{DD}$			1	$\mu\text{A}$
			$V_i = AV_{REF0}$ , $AV_{REF0} = V_{DD} < 2.7\text{ V}$				
	I <sub>LIH3</sub>	P110, P111	$V_i = AV_{REF1}$ , $2.7\text{ V} \leq AV_{REF1} \leq V_{DD}$				1
$V_i = AV_{REF1}$ , $AV_{REF1} = V_{DD} < 2.7\text{ V}$							
I <sub>LIH4</sub>	P121 to P124 (X1, X2, XT1, XT2)	$V_i = V_{DD}$	In Input port			1	$\mu\text{A}$
			In resonator connection			10	$\mu\text{A}$
Input leakage current, low	I <sub>LIL1</sub>	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P131, P140 to P145, P160 to P163, FLMD0, $\overline{\text{RESET}}$	$V_i = V_{SS}$			-1	$\mu\text{A}$
	I <sub>LIL2</sub>	P20 to P27, P150 to P157	$V_i = V_{SS}$ , $2.7\text{ V} \leq AV_{REF0} \leq V_{DD}$			-1	$\mu\text{A}$
			$V_i = V_{SS}$ , $AV_{REF0} = V_{DD} < 2.7\text{ V}$				
	I <sub>LIL3</sub>	P110, P111	$V_i = V_{SS}$ , $2.7\text{ V} \leq AV_{REF1} \leq V_{DD}$				-1
$V_i = V_{SS}$ , $AV_{REF1} = V_{DD} < 2.7\text{ V}$							
I <sub>LIL4</sub>	P121 to P124 (X1, X2, XT1, XT2)	$V_i = V_{SS}$	In Input port			-1	$\mu\text{A}$
			In resonator connection			-10	$\mu\text{A}$

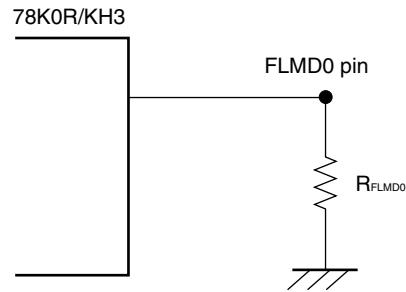
**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**DC Characteristics (7/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
On-chip pull-up resistance	$R_U$	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P131, P140 to P145, P160 to P163 $V_I = V_{SS}$ , In input port	10	20	100	$\text{k}\Omega$
FLMD0 pin external pull-down resistance <small>Note</small>	$R_{FLMD0}$	When enabling the self-programming mode setting with software	100			$\text{k}\Omega$

**Note** It is recommended to leave the FLMD0 pin open. If the pin is required to be pulled down externally, set  $R_{FLMD0}$  to 100  $\text{k}\Omega$  or more.



**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

## DC Characteristics (8/12)

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit		
Supply current	$I_{DD1}$ <sup>Note 1</sup>	Operating mode	$f_{MX} = 20\text{ MHz}$ <sup>Note 2</sup> , $V_{DD} = 5.0\text{ V}$	Square wave input		7.0	12.2	mA	
				Resonator connection		7.3	12.5	mA	
			$f_{MX} = 20\text{ MHz}$ <sup>Note 2</sup> , $V_{DD} = 3.0\text{ V}$	Square wave input		7.0	12.2	mA	
				Resonator connection		7.3	12.5	mA	
			$f_{MX} = 10\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 5.0\text{ V}$	Square wave input		3.8	6.2	mA	
				Resonator connection		3.9	6.3	mA	
			$f_{MX} = 10\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 3.0\text{ V}$	Square wave input		3.8	6.2	mA	
				Resonator connection		3.9	6.3	mA	
			$f_{MX} = 5\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 3.0\text{ V}$	Normal current mode	Square wave input		2.1	3.0	mA
					Resonator connection		2.2	3.1	mA
				Low consumption current mode <sup>Note 4</sup>	Square wave input		1.5	2.1	mA
					Resonator connection		1.5	2.1	mA
			$f_{MX} = 5\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 2.0\text{ V}$	Normal current mode	Square wave input		1.4	2.1	mA
					Resonator connection		1.4	2.1	mA
Low consumption current mode <sup>Note 4</sup>	Square wave input			1.4	2.0	mA			
	Resonator connection			1.4	2.0	mA			
$f_{IH} = 8\text{ MHz}$ <sup>Note 5</sup>		$V_{DD} = 5.0\text{ V}$		3.1	5.0	mA			
		$V_{DD} = 3.0\text{ V}$		3.1	5.0	mA			

**Notes 1.** Total current flowing into  $V_{DD}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $AV_{REF0}$ , and  $AV_{REF1}$ , 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, D/A converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors.

**2.** When internal high-speed oscillator and subsystem clock are stopped.

**3.** When AMPH (bit 0 of clock operation mode control register (CMC)) = 0 and FSEL (bit 0 of operation speed mode control register (OSMC)) = 0.

**4.** When the RMC register is set to 5AH.

**5.** When high-speed system clock and subsystem clock are stopped. When FSEL (bit 0 of operation speed mode control register (OSMC)) = 0 is set.

**Remarks 1.**  $f_{MX}$ : High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

$f_{IH}$ : Internal high-speed oscillation clock frequency

**2.** For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 23 REGULATOR**.

<R> **3.** Temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$

**DC Characteristics (9/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	$I_{DD1}$ <sup>Note 1</sup>	Operating mode	$f_{SUB} = 32.768\text{ kHz}$ <sup>Note 2</sup> , $T_A = -40$ to $+70\text{ }^\circ\text{C}$	$V_{DD} = 5.0\text{ V}$		6.4	36.0	$\mu\text{A}$
				$V_{DD} = 3.0\text{ V}$		6.4	36.0	$\mu\text{A}$
				$V_{DD} = 2.0\text{ V}$		6.3	32.8	$\mu\text{A}$
			$f_{SUB} = 32.768\text{ kHz}$ <sup>Note 2</sup> , $T_A = -40$ to $+85\text{ }^\circ\text{C}$	$V_{DD} = 5.0\text{ V}$		6.4	51.0	$\mu\text{A}$
				$V_{DD} = 3.0\text{ V}$		6.4	51.0	$\mu\text{A}$
				$V_{DD} = 2.0\text{ V}$		6.3	47.8	$\mu\text{A}$

**Notes** 1. Total current flowing into  $V_{DD}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $AV_{REF0}$ , and  $AV_{REF1}$ , 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, D/A converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors.

2. When internal high-speed oscillator and high-speed system clock are stopped. When watchdog timer is stopped.

**Remarks** 1.  $f_{SUB}$ : Subsystem clock frequency (XT1 clock oscillation frequency)

2. Temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$

<R>

**DC Characteristics (10/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit		
Supply current	$I_{DD2}$ <sup>Note 1</sup>	HALT mode	$f_{MX} = 20\text{ MHz}$ <sup>Note 2</sup> , $V_{DD} = 5.0\text{ V}$	Square wave input		1.0	2.7	mA	
				Resonator connection		1.3	3.0	mA	
			$f_{MX} = 20\text{ MHz}$ <sup>Note 2</sup> , $V_{DD} = 3.0\text{ V}$	Square wave input		1.0	2.7	mA	
				Resonator connection		1.3	3.0	mA	
			$f_{MX} = 10\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 5.0\text{ V}$	Square wave input		0.52	1.4	mA	
				Resonator connection		0.62	1.5	mA	
			$f_{MX} = 10\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 3.0\text{ V}$	Square wave input		0.52	1.4	mA	
				Resonator connection		0.62	1.5	mA	
			$f_{MX} = 5\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 3.0\text{ V}$	Normal current mode	Square wave input		0.36	0.75	mA
					Resonator connection		0.41	0.8	mA
				Low consumption current mode <sup>Note 4</sup>	Square wave input		0.22	0.5	mA
					Resonator connection		0.27	0.55	mA
			$f_{MX} = 5\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 2.0\text{ V}$	Normal current mode	Square wave input		0.22	0.5	mA
					Resonator connection		0.27	0.55	mA
Low consumption current mode <sup>Note 4</sup>	Square wave input			0.22	0.5	mA			
	Resonator connection			0.27	0.55	mA			
$f_{IH} = 8\text{ MHz}$ <sup>Note 5</sup>		$V_{DD} = 5.0\text{ V}$		0.45	1.2	mA			
		$V_{DD} = 3.0\text{ V}$		0.45	1.2	mA			

**Notes 1.** Total current flowing into  $V_{DD}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $AV_{REF0}$ , and  $AV_{REF1}$ , including the input leakage current flowing when the level of the input pin is fixed to  $V_{DD}$  or  $V_{SS}$ . The maximum value include the peripheral operation current. However, not including the current flowing into the A/D converter, D/A converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors. During HALT instruction execution by flash memory.

**2.** When internal high-speed oscillator and subsystem clock are stopped.

**3.** When AMPH (bit 0 of clock operation mode control register (CMC)) = 0 and FSEL (bit 0 of operation speed mode control register (OSMC)) = 0.

**4.** When the RMC register is set to 5AH.

**5.** When high-speed system clock and subsystem clock are stopped. When FSEL (bit 0 of operation speed mode control register (OSMC)) = 0 is set.

**Remarks 1.**  $f_{MX}$ : High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

$f_{IH}$ : Internal high-speed oscillation clock frequency

**2.** For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 23 REGULATOR**.

<R> **3.** Temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$

**DC Characteristics (11/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	
Supply current	$I_{DD2}$ <sup>Note 1</sup>	HALT mode	$f_{SUB} = 32.768\text{ kHz}$ <sup>Note 2</sup> , $T_A = -40$ to $+70\text{ }^\circ\text{C}$	$V_{DD} = 5.0\text{ V}$		2.2	26.0	$\mu\text{A}$
				$V_{DD} = 3.0\text{ V}$		2.2	26.0	$\mu\text{A}$
				$V_{DD} = 2.0\text{ V}$		2.1	25.8	$\mu\text{A}$
			$f_{SUB} = 32.768\text{ kHz}$ <sup>Note 2</sup> , $T_A = -40$ to $+85\text{ }^\circ\text{C}$	$V_{DD} = 5.0\text{ V}$		2.2	41.0	$\mu\text{A}$
				$V_{DD} = 3.0\text{ V}$		2.2	41.0	$\mu\text{A}$
				$V_{DD} = 2.0\text{ V}$		2.1	40.8	$\mu\text{A}$
	$I_{DD3}$ <sup>Note 3</sup>	STOP mode	$T_A = -40$ to $+70\text{ }^\circ\text{C}$		1.1	21.0	$\mu\text{A}$	
			$T_A = -40$ to $+85\text{ }^\circ\text{C}$		1.1	36.0	$\mu\text{A}$	

- Notes**
1. Total current flowing into  $V_{DD}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $AV_{REF0}$ , and  $AV_{REF1}$ , including the input leakage current flowing when the level of the input pin is fixed to  $V_{DD}$  or  $V_{SS}$ . The maximum value include the peripheral operation current. However, not including the current flowing into the A/D converter, D/A converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors. During HALT instruction execution by flash memory.
  2. When internal high-speed oscillator and high-speed system clock are stopped. When watchdog timer is stopped.
  3. Total current flowing into  $V_{DD}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $AV_{REF0}$ , and  $AV_{REF1}$ , including the input leakage current flowing when the level of the input pin is fixed to  $V_{DD}$  or  $V_{SS}$ . When subsystem clock is stopped. When watchdog timer is stopped.

- Remarks**
1.  $f_{SUB}$ : Subsystem clock frequency (XT1 clock oscillation frequency)
  2. Temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$

&lt;R&gt;

**DC Characteristics (12/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
RTC operating current	I <sub>RTC</sub> <sup>Notes 1, 2</sup>	f <sub>SUB</sub> = 32.768 kHz	V <sub>DD</sub> = 3.0 V		0.2	1.0	μA
			V <sub>DD</sub> = 2.0 V		0.2	1.0	
Watchdog timer operating current	I <sub>WDT</sub> <sup>Notes 2, 3</sup>	f <sub>IL</sub> = 240 kHz		5	10	μA	
A/D converter operating current	I <sub>ADC</sub> <sup>Note 4</sup>	During conversion at maximum speed, 2.3 V ≤ AV <sub>REF0</sub>		0.86	1.9	mA	
D/A converter operating current	I <sub>DAC</sub> <sup>Note 5</sup>	Per 1 channel		1.0	2.5	mA	
LVI operating current	I <sub>LVI</sub> <sup>Note 6</sup>			9	18	μA	

- Notes**
- Current flowing only to the real-time counter (excluding the operating current of the XT1 oscillator). The current value of the 78K0R/KH3 is the TYP. value, the sum of the TYP. values of either I<sub>DD1</sub> or I<sub>DD2</sub>, and I<sub>RTC</sub>, when the real-time counter operates in operation mode or HALT mode. The I<sub>DD1</sub> and I<sub>DD2</sub> MAX. values also include the real-time counter operating current.
  - When internal high-speed oscillator and high-speed system clock are stopped.
  - Current flowing only to the watchdog timer (including the operating current of the 240 kHz internal oscillator). The current value of the 78K0R/KH3 is the sum of I<sub>DD1</sub>, I<sub>DD2</sub> or I<sub>DD3</sub> and I<sub>WDT</sub> when f<sub>CLK</sub> = f<sub>SUB</sub>/2 or when the watchdog timer operates in STOP mode.
  - Current flowing only to the A/D converter (AV<sub>REF0</sub> pin). The current value of the 78K0R/KH3 is the sum of I<sub>DD1</sub> or I<sub>DD2</sub> and I<sub>ADC</sub> when the A/D converter operates in an operation mode or the HALT mode.
  - Current flowing only to the D/A converter (AV<sub>REF1</sub> pin). The current value of the 78K0R/KH3 is the sum of I<sub>DD1</sub> or I<sub>DD2</sub> and I<sub>DAC</sub> when the D/A converter operates in an operation mode or the HALT mode.
  - Current flowing only to the LVI circuit. The current value of the 78K0R/KH3 is the sum of I<sub>DD1</sub>, I<sub>DD2</sub> or I<sub>DD3</sub> and I<sub>LVI</sub> when the LVI circuit operates in the Operating, HALT or STOP mode.

- Remarks**
- f<sub>IL</sub>: Internal low-speed oscillation clock frequency  
f<sub>SUB</sub>: Subsystem clock frequency (XT1 clock oscillation frequency)  
f<sub>CLK</sub>: CPU/peripheral hardware clock frequency

<R> 2. Temperature condition of the TYP. value is T<sub>A</sub> = 25°C



## AC Characteristics

## (1) Basic operation (1/6)

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	
Instruction cycle (minimum instruction execution time)	$t_{CY}$	Main system clock ( $f_{MAIN}$ ) operation	Normal current mode	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0.05		8	$\mu\text{s}$
				$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	0.2		8	$\mu\text{s}$
			Low consumption current mode	0.2		8	$\mu\text{s}$	
		Subsystem clock ( $f_{SUB}$ ) operation			57.2	61	62.5	$\mu\text{s}$
		In the self programming mode	Normal current mode	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0.05		0.5	$\mu\text{s}$
External main system clock frequency	$f_{EX}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	Normal current mode	2.0		20.0	MHz	
			Low consumption current mode	2.0		5.0	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		2.0		5.0	MHz	
External main system clock input high-level width, low-level width	$t_{EXH}$ , $t_{EXL}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	Normal current mode	24			ns	
			Low consumption current mode	96			ns	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		96			ns	
TI00 to TI07 input high-level width, low- level width	$t_{TIH}$ , $t_{TIL}$			$1/f_{MCK} + 10$			ns	
TO00 to TO07 output frequency	$f_{TO}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$				10	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$				5	MHz	
PCLBUZ0, PCLBUZ1 output frequency	$f_{PCL}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$				10	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$				5	MHz	
Interrupt input high- level width, low-level width	$t_{INTH}$ , $t_{INTL}$			1			$\mu\text{s}$	
Key interrupt input low-level width	$t_{KR}$			250			ns	
$\overline{\text{RESET}}$ low-level width	$t_{RSL}$			10			$\mu\text{s}$	

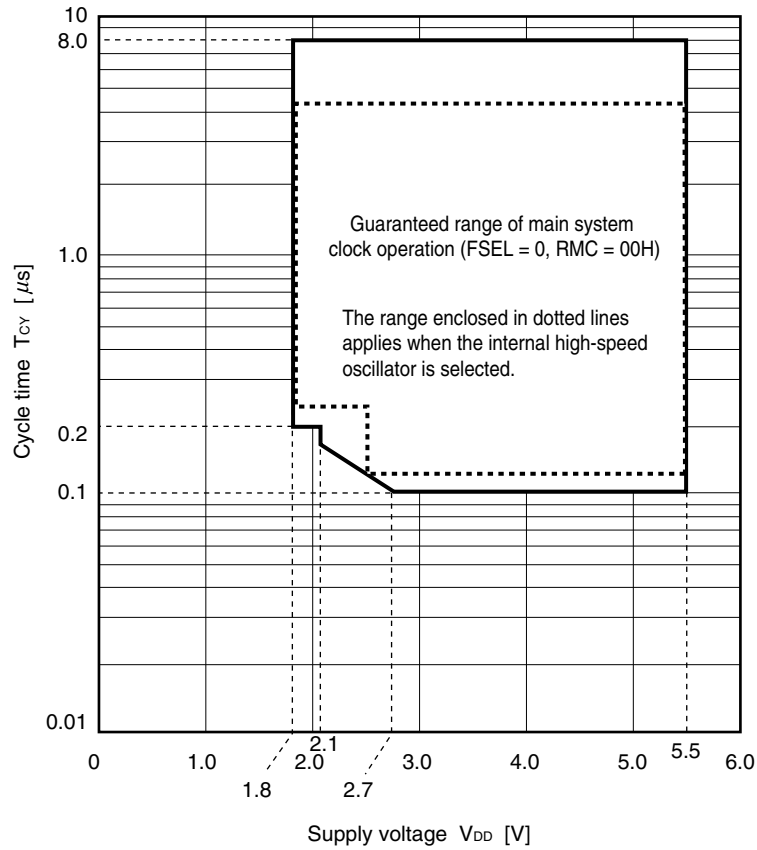
**Remarks 1.**  $f_{MCK}$ : Timer array unit operation clock frequency

(Operation clock to be set by the CKS0n bit of the TMR0n register. n: Channel number (n = 0 to 7))

- For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 23 REGULATOR**.

## (1) Basic operation (2/6)

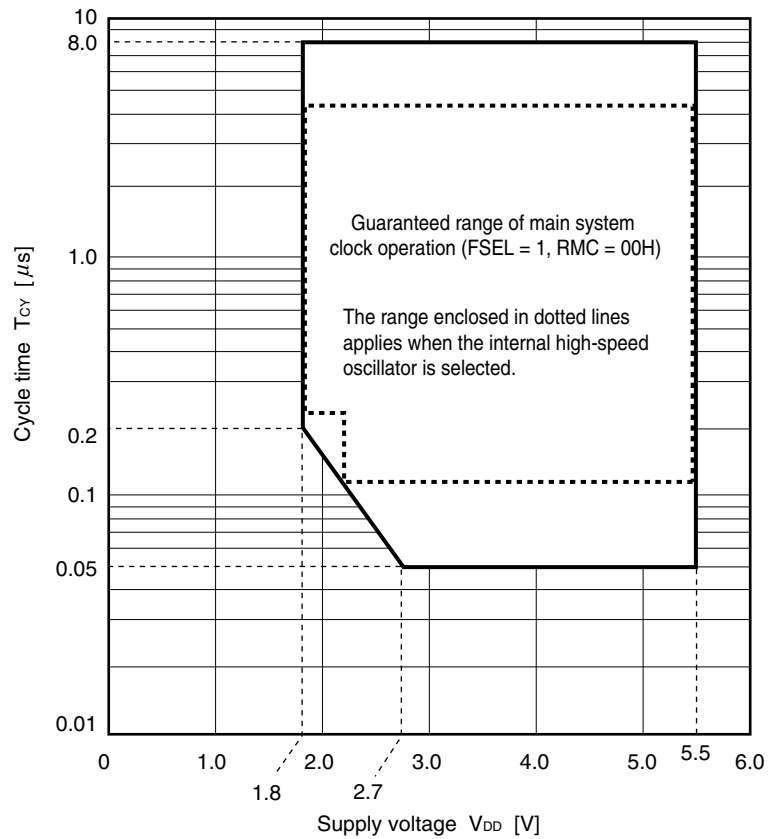
## Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 00H)



**Remark** FSEL: Bit 0 of the operation speed mode control register (OSMC)  
RMC: Regulator mode control register

## (1) Basic operation (3/6)

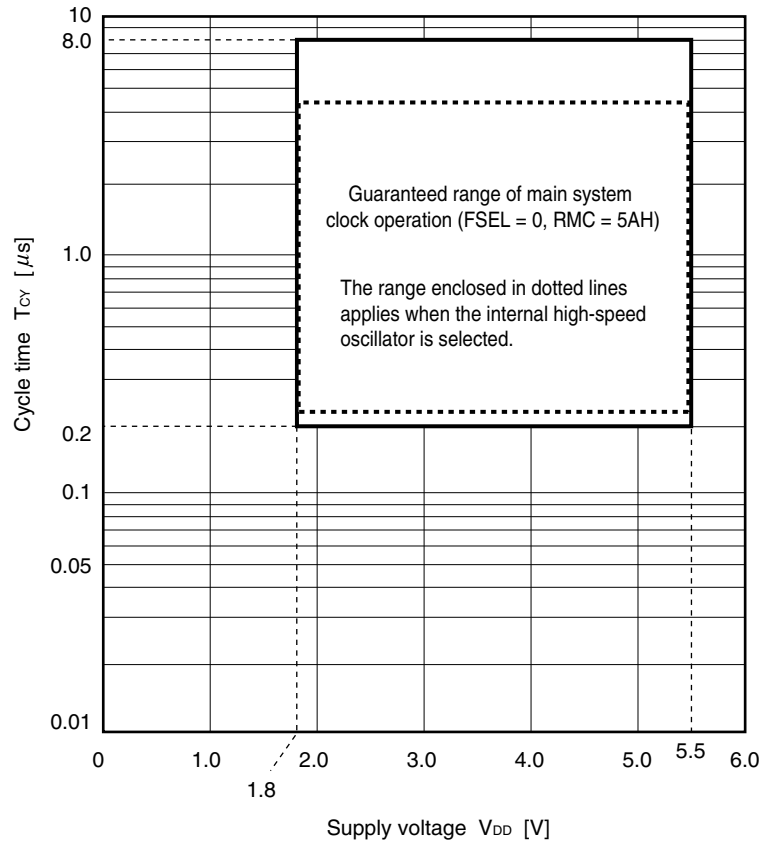
## Minimum instruction execution time during main system clock operation (FSEL = 1, RMC = 00H)



**Remark** FSEL: Bit 0 of the operation speed mode control register (OSMC)  
 RMC: Regulator mode control register

## (1) Basic operation (4/6)

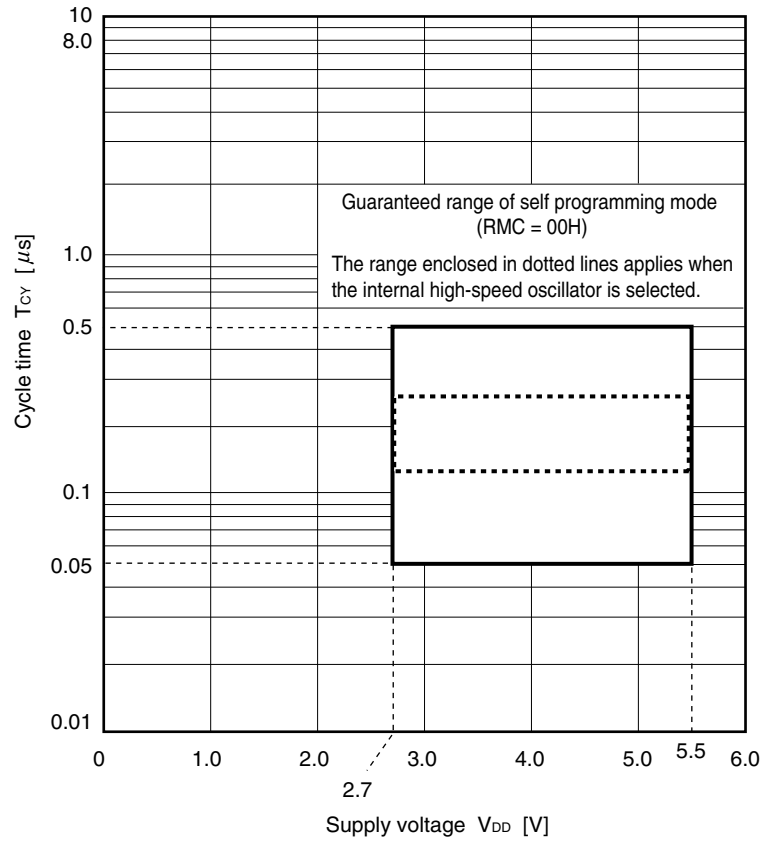
## Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 5AH)



- Remarks**
1. FSEL: Bit 0 of the operation speed mode control register (OSMC)  
RMC: Regulator mode control register
  2. The entire voltage range is 5 MHz (MAX.) when RMC is set to 5AH.

## (1) Basic operation (5/6)

## Minimum instruction execution time during self programming mode (RMC = 00H)

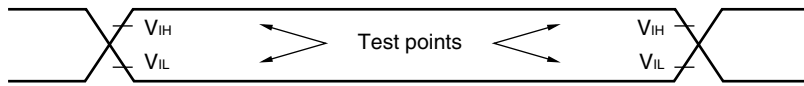


**Remarks 1.** RMC: Regulator mode control register

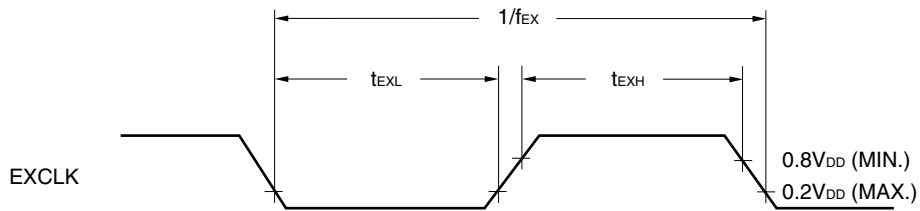
**2.** The self programming function cannot be used when RMC is set to 5AH or the CPU operates with the subsystem clock.

(1) Basic operation (6/6)

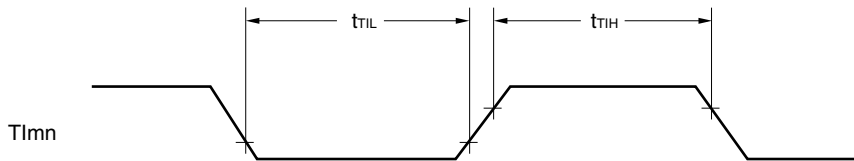
AC Timing Test Points (Excluding external bus interface)



External Main System Clock Timing

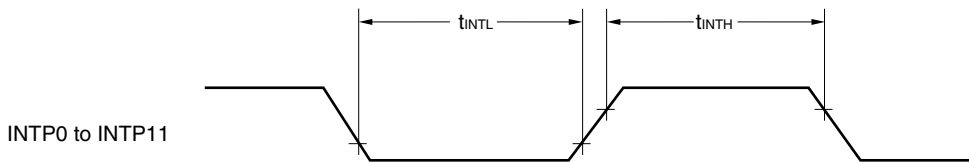


TI Timing

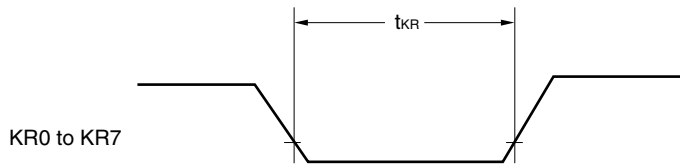


**Remark** n: Channel number, m: Unit number, mn = 00 to 07, 10 to 13

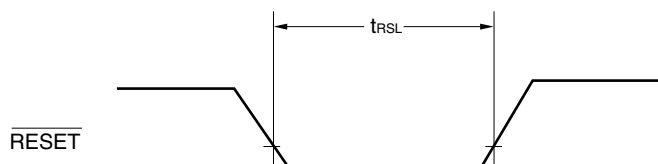
Interrupt Request Input Timing



Key Interrupt Input Timing



RESET Input Timing



## (2) External bus interface (1/5)

## (a) Read/write cycle (CLKOUT synchronous)

• Conventional-specification products ( $\mu$ PD78F117x)(T<sub>A</sub> = -40 to +85°C, 2.7 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CLKOUT cycle	t <sub>CYK</sub>	<1> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	100			ns
CLKOUT high-level width	t <sub>WKH</sub>	<2> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.4t <sub>CYK</sub> - 30		0.6t <sub>CYK</sub>	ns
CLKOUT low-level width	t <sub>WKL</sub>	<3> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.4t <sub>CYK</sub> - 30		0.6t <sub>CYK</sub>	ns
ASTB high-level width	t <sub>WASH1</sub>	<4> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.8t <sub>CYK</sub> - 40		1.2t <sub>CYK</sub>	ns
$\overline{\text{RD}}$ low-level width	t <sub>WRDL1</sub>	<5> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	(0.8 + m + w) t <sub>CYK</sub> - 40		(1.2 + m + w) t <sub>CYK</sub>	ns
$\overline{\text{WR0}}$ , $\overline{\text{WR1}}$ low-level width	t <sub>WWRL1</sub>	<6> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	(0.8 + w) t <sub>CYK</sub> - 40		(1.2 + w) t <sub>CYK</sub>	ns
Delay time from CLKOUT↑ to ASTB	t <sub>DKAS1</sub>	<7> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2		50	ns
Delay time from CLKOUT↑ to $\overline{\text{RD}}$	t <sub>DKRD1</sub>	<8> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2		50	ns
Delay time from CLKOUT↑ to $\overline{\text{WR0}}$ , $\overline{\text{WR1}}$	t <sub>DKWR1</sub>	<9> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2		50	ns
Delay time from CLKOUT↑ to address	t <sub>DKA1</sub>	<10> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2		55	ns
Data output delay time from CLKOUT↑	t <sub>DKOD1</sub>	<11> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2		55	ns
Data output hold time from CLKOUT↑	t <sub>HKOD1</sub>	<12> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2			ns
Data input setup time to CLKOUT↑	t <sub>SKDH1</sub>	<13> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	40			ns
Data input hold time from CLKOUT↑	t <sub>HKDH1</sub>	<14> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0			ns
$\overline{\text{WAIT}}$ setup time to CLKOUT↑	t <sub>SKWT1</sub>	<15> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	45			ns
$\overline{\text{WAIT}}$ hold time from CLKOUT↑	t <sub>HKWT1</sub>	<16> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0			ns
Delay time from address output stop to $\overline{\text{RD}}$ ↓	t <sub>DAR1</sub>	<17> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0			ns

**Remarks 1.** CL: The pin load capacitance is 15 pF.**2.** Test points: V<sub>OH</sub> = 0.8V<sub>DD</sub>, V<sub>OL</sub> = 0.2V<sub>DD</sub>**3.** m = 0: Multiplexed bus mode

m = 1: Separate bus mode

w: Number of waits with  $\overline{\text{WAIT}}$

## (2) External bus interface (2/5)

• Expanded-specification products ( $\mu$ PD78F117xA) $(T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

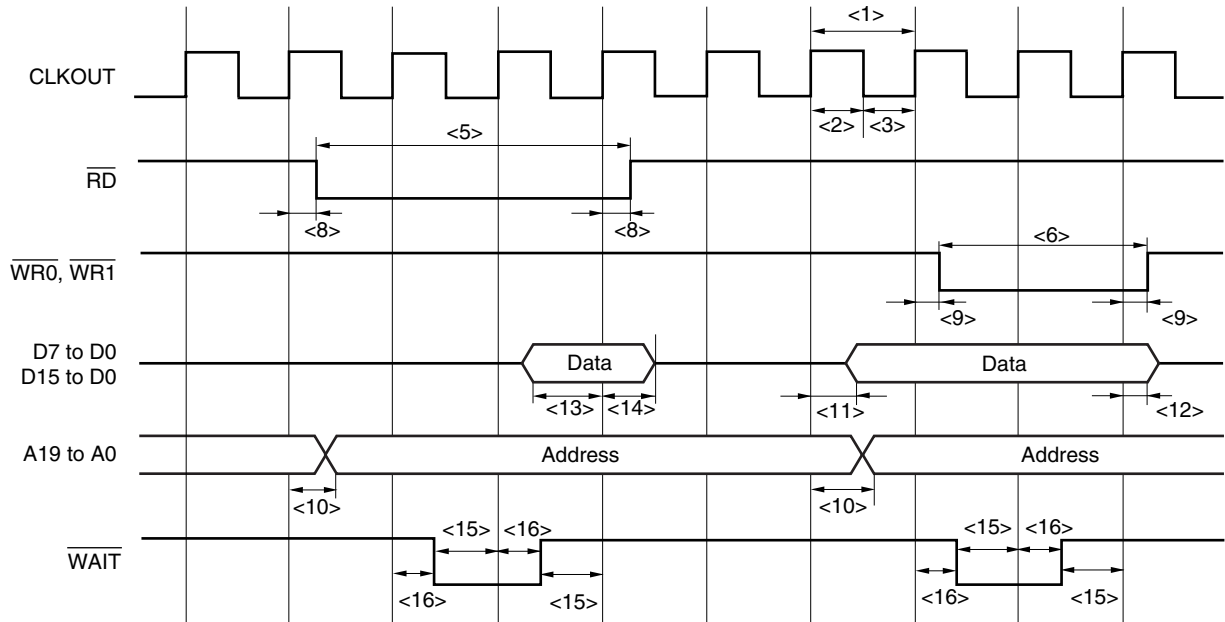
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CLKOUT cycle	$t_{CYK}$	<1>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	100		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	200		ns
CLKOUT high-level width	$t_{WKH}$	<2>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$0.4t_{CYK} - 30$		$0.6t_{CYK}$
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$0.4t_{CYK} - 50$		$0.6t_{CYK}$
CLKOUT low-level width	$t_{WKL}$	<3>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$0.4t_{CYK} - 30$		$0.6t_{CYK}$
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$0.4t_{CYK} - 50$		$0.6t_{CYK}$
ASTB high-level width	$t_{WASH1}$	<4>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$0.8t_{CYK} - 40$		$1.2t_{CYK}$
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$0.8t_{CYK} - 60$		$1.2t_{CYK}$
$\overline{\text{RD}}$ low-level width	$t_{WRDL1}$	<5>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$(0.8 + m + w)t_{CYK} - 40$		$(1.2 + m + w)t_{CYK}$
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$(0.8 + m + w)t_{CYK} - 60$		$(1.2 + m + w)t_{CYK}$
$\overline{\text{WR0}}$ , $\overline{\text{WR1}}$ low-level width	$t_{WWRL1}$	<6>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$(0.8 + w)t_{CYK} - 40$		$(1.2 + w)t_{CYK}$
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$(0.8 + w)t_{CYK} - 60$		$(1.2 + w)t_{CYK}$
Delay time from CLKOUT $\uparrow$ to ASTB	$t_{DKAS1}$	<7>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2		50
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	2		80
Delay time from CLKOUT $\uparrow$ to $\overline{\text{RD}}$	$t_{DKRD1}$	<8>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2		50
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	2		80
Delay time from CLKOUT $\uparrow$ to $\overline{\text{WR0}}$ , $\overline{\text{WR1}}$	$t_{DKWR1}$	<9>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2		50
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	2		80
Delay time from CLKOUT $\uparrow$ to address	$t_{DKA1}$	<10>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2		55
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	2		90
Data output delay time from CLKOUT $\uparrow$	$t_{DKOD1}$	<11>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2		55
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	2		90
Data output hold time from CLKOUT $\uparrow$	$t_{HKOD1}$	<12>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	2		ns
Data input setup time to CLKOUT $\uparrow$	$t_{SKDH1}$	<13>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	40		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	85		ns
Data input hold time from CLKOUT $\uparrow$	$t_{HKDH1}$	<14>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	0		ns
$\overline{\text{WAIT}}$ setup time to CLKOUT $\uparrow$	$t_{SKWT1}$	<15>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	45		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	85		ns
$\overline{\text{WAIT}}$ hold time from CLKOUT $\uparrow$	$t_{HKWT1}$	<16>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	0		ns
Delay time from address output stop to $\overline{\text{RD}}\downarrow$	$t_{DAR1}$	<17>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	0		ns

**Remarks 1.** CL: The pin load capacitance is 15 pF.**2.** Test points:  $V_{OH} = 0.8V_{DD}$ ,  $V_{OL} = 0.2V_{DD}$ **3.**  $m = 0$ : Multiplexed bus mode $m = 1$ : Separate bus modew: Number of waits with  $\overline{\text{WAIT}}$

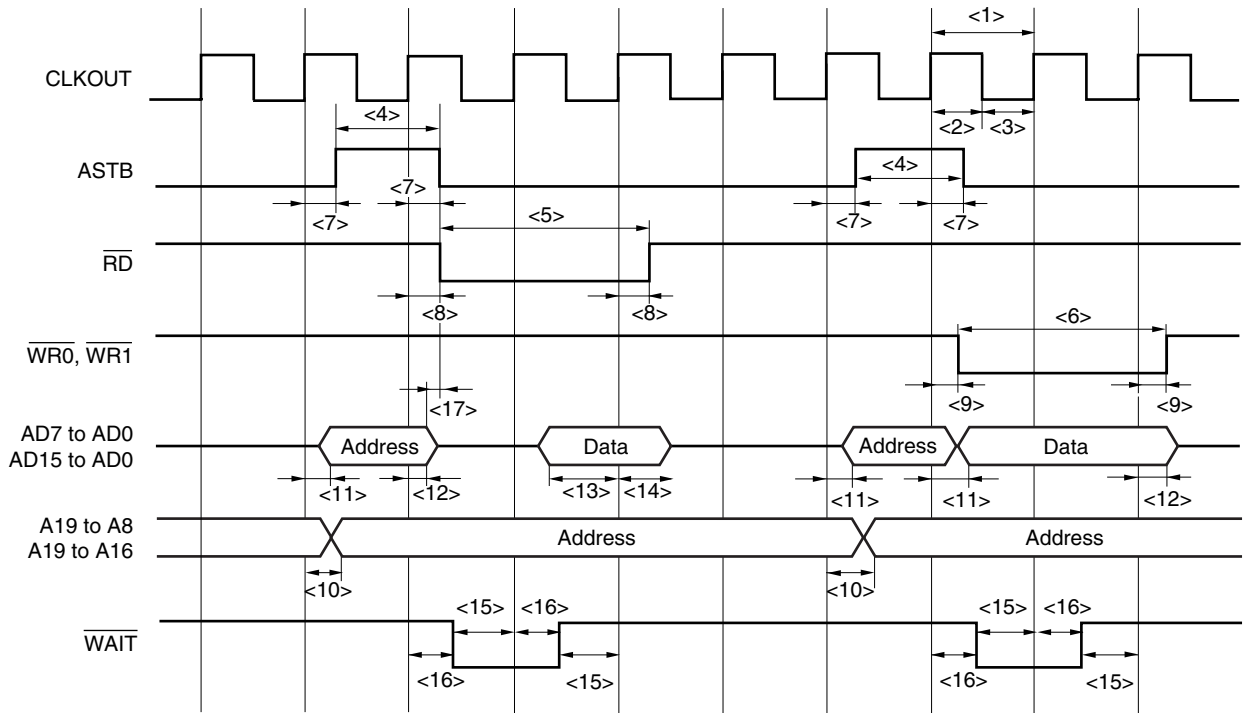


(2) External bus interface (3/5)

Read/write cycle (CLKOUT synchronous): In separate bus mode



Read/write cycle (CLKOUT synchronous): In multiplexed bus mode



## (2) External bus interface (4/5)

## (b) Read/write cycle (CLKOUT asynchronous)

• Conventional-specification products ( $\mu$ PD78F117x)(T<sub>A</sub> = -40 to +85°C, 2.7 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CLKOUT cycle	t <sub>CYK2</sub>	<18> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	100			ns
$\overline{RD}$ low-level width	t <sub>WRDL2</sub>	<19> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	1.8t <sub>CYK2</sub> - 40		2.2t <sub>CYK2</sub>	ns
$\overline{WR0}$ , $\overline{WR1}$ low-level width	t <sub>WWRL2</sub>	<20> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.8t <sub>CYK2</sub> - 40		1.2t <sub>CYK2</sub>	ns
Data input setup time to $\overline{RD}\uparrow$	t <sub>SRDDI2</sub>	<21> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	90			ns
Data input hold time from $\overline{RD}\uparrow$	t <sub>HRDDI2</sub>	<22> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0			ns
Data output setup time to $\overline{WR0}$ , $\overline{WR1}\downarrow$	t <sub>SWROD2</sub>	<23> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	t <sub>CYK2</sub> - 5			ns
Data output hold time from $\overline{WR0}$ , $\overline{WR1}\uparrow$	t <sub>HKOD2</sub>	<24> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2			ns
Delay time from $\overline{RD}\downarrow$ to address	t <sub>DRDA2</sub>	<25> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V			5	ns
Address setup time to $\overline{WR0}$ , $\overline{WR1}\downarrow$	t <sub>SWRA2</sub>	<26> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	t <sub>CYK2</sub> - 5			ns

• Expanded-specification products ( $\mu$ PD78F117xA)(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)

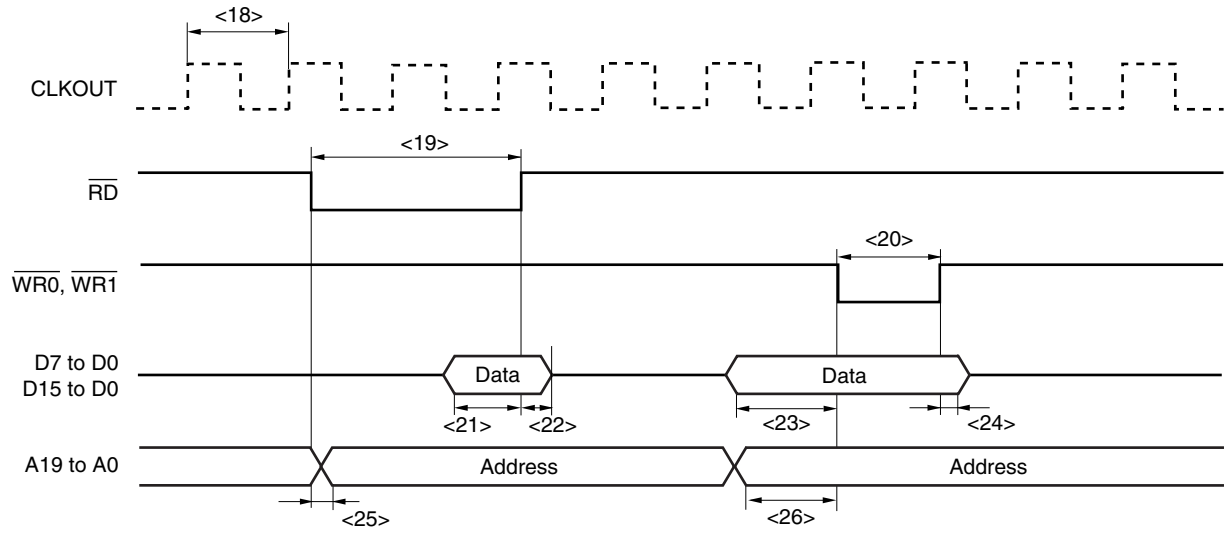
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CLKOUT cycle	t <sub>CYK2</sub>	<18> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	100			ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V	200			ns
$\overline{RD}$ low-level width	t <sub>WRDL2</sub>	<19> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	1.8t <sub>CYK2</sub> - 40		2.2t <sub>CYK2</sub>	ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V	1.8t <sub>CYK2</sub> - 60		2.2t <sub>CYK2</sub>	ns
$\overline{WR0}$ , $\overline{WR1}$ low-level width	t <sub>WWRL2</sub>	<20> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0.8t <sub>CYK2</sub> - 40		1.2t <sub>CYK2</sub>	ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V	0.8t <sub>CYK2</sub> - 60		1.2t <sub>CYK2</sub>	ns
Data input setup time to $\overline{RD}\uparrow$	t <sub>SRDDI2</sub>	<21> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	90			ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V	170			ns
Data input hold time from $\overline{RD}\uparrow$	t <sub>HRDDI2</sub>	<22> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	0			ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V	0			ns
Data output setup time to $\overline{WR0}$ , $\overline{WR1}\downarrow$	t <sub>SWROD2</sub>	<23> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	t <sub>CYK2</sub> - 5			ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V	t <sub>CYK2</sub> - 15			ns
Data output hold time from $\overline{WR0}$ , $\overline{WR1}\uparrow$	t <sub>HKOD2</sub>	<24> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	2			ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V	2			ns
Delay time from $\overline{RD}\downarrow$ to address	t <sub>DRDA2</sub>	<25> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V			5	ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V			15	ns
Address setup time to $\overline{WR0}$ , $\overline{WR1}\downarrow$	t <sub>SWRA2</sub>	<26> 2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V	t <sub>CYK2</sub> - 5			ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V	t <sub>CYK2</sub> - 15			ns

- Cautions**
1. CLKOUT output is not used during CLKOUT asynchronous operation, but a CPU wait occurs according to the setting of bits 4 and 5 (EW0, EW1) of the memory expansion mode control register (MEM). When f<sub>CLK</sub> is sufficiently high, insert a wait by setting the EW0 and EW1 bits.
  2. Do not use the WAIT pin during CLKOUT asynchronous operation.  
Use the separate bus mode during CLKOUT asynchronous operation.

- Remarks**
1. f<sub>CLK</sub>: CPU/peripheral hardware clock frequency
  2. CL: The pin load capacitance is 15 pF.
  3. Test points: V<sub>OH</sub> = 0.8V<sub>DD</sub>, V<sub>OL</sub> = 0.2V<sub>DD</sub>

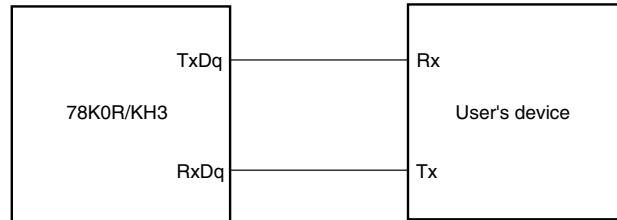
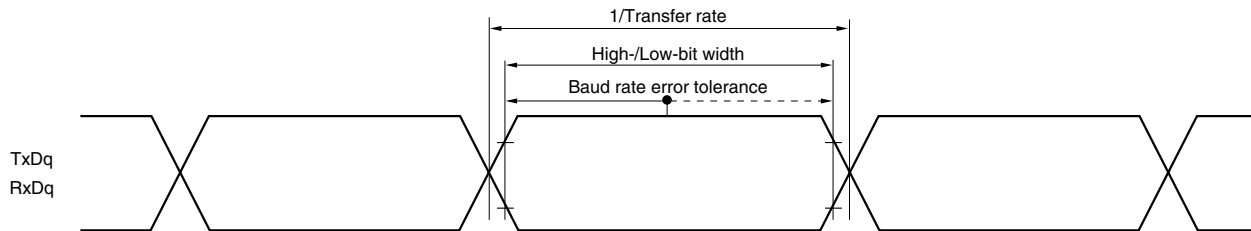
(2) External bus interface (5/5)

Read/write cycle (CLKOUT asynchronous): In separate bus mode



**(3) Serial interface: Serial array unit (1/18)** $(T_A = -40 \text{ to } +85^\circ\text{C}, 1.8 \text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5 \text{ V}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 \text{ V})$ **(a) During communication at same potential (UART mode) (dedicated baud rate generator output)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					$f_{MCK}/6$	bps
		$f_{CLK} = 20 \text{ MHz}, f_{MCK} = f_{CLK}$			3.3	Mbps

**UART mode connection diagram (during communication at same potential)****UART mode bit width (during communication at same potential) (reference)**

**Caution** Select the normal input buffer for RxDi and the normal output mode for TxDi by using the PIMg and POMg registers.

- Remarks**
- q: UART number (q = 0 to 3), g: PIM and POM number (g = 0, 14),  
i: UART number for which communication at different potential can be selected (i = 1, 2)
  - $f_{MCK}$ : Serial array unit operation clock frequency  
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1),  
n: Channel number (n = 0 to 3))

## (3) Serial interface: Serial array unit (2/18)

(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)

&lt;R&gt;

(b) During communication at same potential (CSI mode) (master mode,  $\overline{\text{SCKp}}$ ... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t <sub>KCY1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V	200 <sup>Note 1</sup>			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V	300 <sup>Note 1</sup>			ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V	600 <sup>Note 1</sup>			ns
$\overline{\text{SCKp}}$ high-/low-level width	t <sub>KH1</sub> , t <sub>KL1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V	t <sub>KCY1</sub> /2 - 20			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V	t <sub>KCY1</sub> /2 - 35			ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V	t <sub>KCY1</sub> /2 - 80			ns
Slp setup time (to $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 2</sup>	t <sub>SIK1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V	70			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V	100			ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V	190			ns
Slp hold time (from $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 3</sup>	t <sub>KSI1</sub>		30			ns
Delay time from $\overline{\text{SCKp}}\downarrow$ to SOp output <sup>Note 4</sup>	t <sub>KSO1</sub>	C = 30 pF <sup>Note 5</sup>			40	ns

**Notes** 1. The value must also be 4/f<sub>CLK</sub> or more.2. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The Slp setup time becomes “to  $\overline{\text{SCKp}}\downarrow$ ” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.3. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The Slp hold time becomes “from  $\overline{\text{SCKp}}\downarrow$ ” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.4. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The delay time to SOp output becomes “from  $\overline{\text{SCKp}}\uparrow$ ” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.5. C is the load capacitance of the  $\overline{\text{SCKp}}$  and SOp output lines.**Caution** Select the normal input buffer for Slj and the normal output mode for SOj and  $\overline{\text{SCKj}}$  by using the PIMg and POMg registers.

**Remarks** 1. p: CSI number (p = 00, 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14),  
j: CSI number for which communication at different potential can be selected (j = 01, 10, 11, 20, 21)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

**(3) Serial interface: Serial array unit (3/18)**(TA = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)<R> **(c) During communication at same potential (CSI mode) (slave mode,  $\overline{\text{SCKp}}$ ... external clock input)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t <sub>KCY2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V	6/f <sub>MCK</sub>			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V	16 MHz < f <sub>MCK</sub>	8/f <sub>MCK</sub>		ns
			f <sub>MCK</sub> ≤ 16 MHz	6/f <sub>MCK</sub>		ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V	16 MHz < f <sub>MCK</sub>	8/f <sub>MCK</sub>		ns
f <sub>MCK</sub> ≤ 16 MHz	6/f <sub>MCK</sub>			ns		
$\overline{\text{SCKp}}$ high-/low-level width	t <sub>KH2</sub> , t <sub>KL2</sub>		f <sub>KCY2</sub> /2			ns
Slp setup time (to $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 1</sup>	t <sub>SIK2</sub>		80			ns
Slp hold time (from $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 2</sup>	t <sub>KSI2</sub>		1/f <sub>MCK</sub> + 50			ns
Delay time from $\overline{\text{SCKp}}\downarrow$ to SO <sub>p</sub> output <sup>Note 3</sup>	t <sub>KSO2</sub>	C = 30 pF <sup>Note 4</sup>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V		2/f <sub>MCK</sub> + 45	ns
			2.7 V ≤ V <sub>DD</sub> < 4.0 V		2/f <sub>MCK</sub> + 57	ns
			1.8 V ≤ V <sub>DD</sub> < 2.7 V		2/f <sub>MCK</sub> + 125	ns

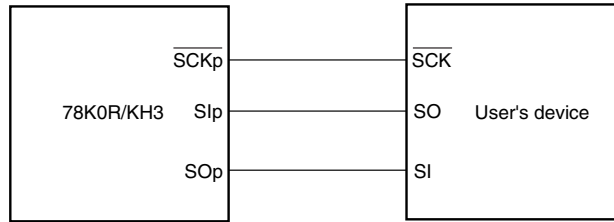
- Notes**
1. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The Slp setup time becomes “to  $\overline{\text{SCKp}}\uparrow$ ” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
  2. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The Slp setup time becomes “from  $\overline{\text{SCKp}}\downarrow$ ” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
  3. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The delay time to SO<sub>p</sub> output becomes “from  $\overline{\text{SCKp}}\downarrow$ ” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
  4. C is the load capacitance of the SO<sub>p</sub> output line.

**Caution** Select the normal input buffer for Sl<sub>j</sub> and  $\overline{\text{SCKj}}$  and the normal output mode for SO<sub>j</sub> by using the PIM<sub>g</sub> and POM<sub>g</sub> registers.

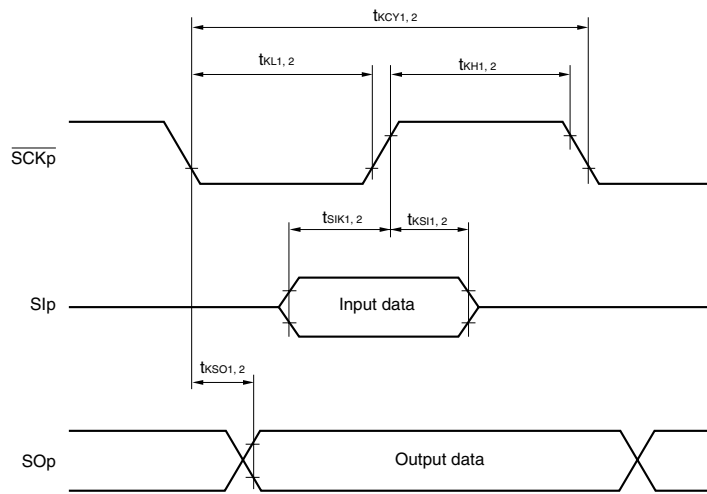
- Remarks**
1. p: CSI number (p = 00, 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14),  
j: CSI number for which communication at different potential can be selected (j = 01, 10, 11, 20, 21)
  2. f<sub>MCK</sub>: Serial array unit operation clock frequency  
(Operation clock to be set by the CKS<sub>mn</sub> bit of the SMR<sub>mn</sub> register. m: Unit number (m = 0, 1),  
n: Channel number (n = 0 to 3))

(3) Serial interface: Serial array unit (4/18)

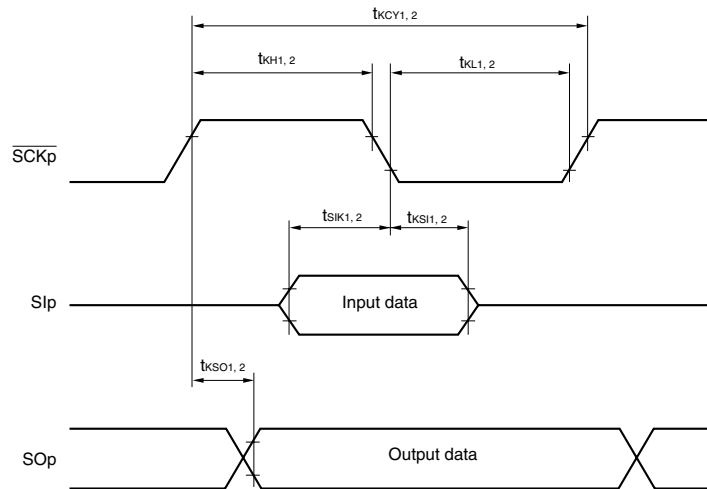
CSI mode connection diagram (during communication at same potential)



CSI mode serial transfer timing (during communication at same potential)  
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



CSI mode serial transfer timing (during communication at same potential)  
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



- Remarks**
1. p: CSI number (p = 00, 01, 10, 11, 20, 21)
  2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

## (3) Serial interface: Serial array unit (5/18)

(d) During communication at same potential (simplified I<sup>2</sup>C mode)• Conventional-specification products ( $\mu$ PD78F117x)(T<sub>A</sub> = -40 to +85°C, 2.7 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
<R> SCLr clock frequency	f <sub>SCL</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ		400 <sup>Note</sup>	kHz
Hold time when SCLr = "L"	t <sub>LOW</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	995		ns
Hold time when SCLr = "H"	t <sub>HIGH</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	995		ns
Data setup time (reception)	t <sub>SU:DAT</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	1/f <sub>MCK</sub> + 120		ns
Data hold time (transmission)	t <sub>HD:DAT</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	0	160	ns

<R> **Note** The value must also be f<sub>MCK</sub>/4 or less.• Expanded-specification products ( $\mu$ PD78F117xA)(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)

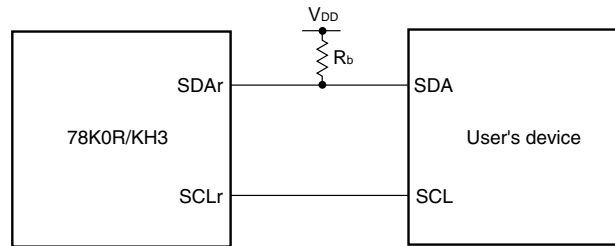
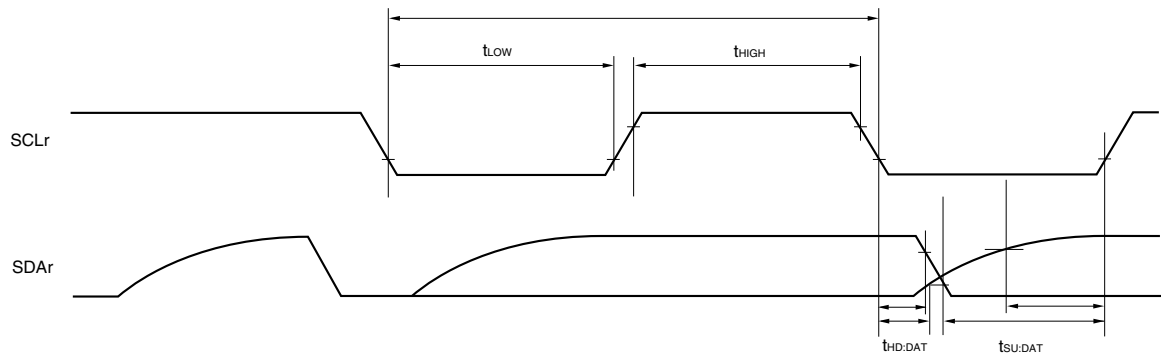
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
<R> SCLr clock frequency	f <sub>SCL</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ		400 <sup>Note</sup>	kHz
		1.8 V ≤ V <sub>DD</sub> < 2.7 V C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ		300 <sup>Note</sup>	kHz
Hold time when SCLr = "L"	t <sub>LOW</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	995		ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	1500		ns
Hold time when SCLr = "H"	t <sub>HIGH</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	995		ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	1500		ns
Data setup time (reception)	t <sub>SU:DAT</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	1/f <sub>MCK</sub> + 120		ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	1/f <sub>MCK</sub> + 230		ns
Data hold time (transmission)	t <sub>HD:DAT</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	0	160	ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	0	210	ns

<R> **Note** The value must also be f<sub>MCK</sub>/4 or less.

(Remarks are given on the next page.)



## (3) Serial interface: Serial array unit (6/18)

Simplified I<sup>2</sup>C mode connection diagram (during communication at same potential)Simplified I<sup>2</sup>C mode serial transfer timing (during communication at same potential)

**Caution** Select the normal input buffer and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for SDAr and the normal output mode for SCLr by using the PIMg and POMg registers.

- Remarks**
1.  $R_b[\Omega]$ : Communication line (SDAr) pull-up resistance,  
 $C_b[F]$ : Communication line (SCLr, SDAr) load capacitance
  2. r: IIC number (r = 10, 11, 20, 21), g: PIM and POM number (g = 0, 9, 12, 14)
  3.  $f_{MCK}$ : Serial array unit operation clock frequency  
 (Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1),  
 n: Channel number (n = 0 to 3), mn = 02, 03, 10, 11)

**(3) Serial interface: Serial array unit (7/18)****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )****(e) During communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output) (1/2)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Transfer rate		Reception	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$			$f_{MCK}/6$	bps
				$f_{CLK} = 20\text{ MHz}$ , $f_{MCK} = f_{CLK}$			3.3
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$ , $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$			$f_{MCK}/6$	bps
				$f_{CLK} = 20\text{ MHz}$ , $f_{MCK} = f_{CLK}$			3.3

**Caution** Select the TTL input buffer for RxDq and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for TxDq by using the PIMg and POMg registers.

**Remarks 1.** q: UART number (q = 1, 2), g: PIM and POM number (g = 0, 14)

**2.**  $f_{MCK}$ : Serial array unit operation clock frequency

(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3))

**3.**  $V_{IH}$  and  $V_{IL}$  below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.

$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$ :  $V_{IH} = 2.2\text{ V}$ ,  $V_{IL} = 0.8\text{ V}$

$2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}$ ,  $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$ :  $V_{IH} = 2.0\text{ V}$ ,  $V_{IL} = 0.5\text{ V}$

**4.** UART0 and UART3 cannot communicate at different potential. Use UART1 and UART2 for communication at different potential.

**(3) Serial interface: Serial array unit (8/18)** $(T_A = -40 \text{ to } +85^\circ\text{C}, 2.7 \text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5 \text{ V}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 \text{ V})$ **(e) During communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output) (2/2)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Transfer rate		Transmission	4.0 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, 2.7 V $\leq$ V <sub>b</sub> $\leq$ 4.0 V			<b>Note 1</b>	
			f <sub>CLK</sub> = 16.8 MHz, f <sub>MCK</sub> = f <sub>CLK</sub> , C <sub>b</sub> = 50 pF, R <sub>b</sub> = 1.4 k $\Omega$ , V <sub>b</sub> = 2.7 V			2.8 <sup>Note 2</sup>	Mbps
			2.7 V $\leq$ V <sub>DD</sub> < 4.0 V, 2.3 V $\leq$ V <sub>b</sub> $\leq$ 2.7 V			<b>Note 3</b>	
			f <sub>CLK</sub> = 19.2 MHz, f <sub>MCK</sub> = f <sub>CLK</sub> , C <sub>b</sub> = 50 pF, R <sub>b</sub> = 2.7 k $\Omega$ , V <sub>b</sub> = 2.3 V			1.2 <sup>Note 4</sup>	Mbps

**Notes 1.** The smaller maximum transfer rate derived by using f<sub>MCK</sub>/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V  $\leq$  V<sub>DD</sub> = EV<sub>DD</sub>  $\leq$  5.5 V and 2.7 V  $\leq$  V<sub>b</sub>  $\leq$  4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

\* This value is the theoretical value of the relative difference between the transmission and reception sides.

2. This value as an example is calculated when the conditions described in the "Conditions" column are met. Refer to Note 1 above to calculate the maximum transfer rate under conditions of the customer.

3. The smaller maximum transfer rate derived by using f<sub>MCK</sub>/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V  $\leq$  V<sub>DD</sub> = EV<sub>DD</sub> < 4.0 V and 2.3 V  $\leq$  V<sub>b</sub>  $\leq$  2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

\* This value is the theoretical value of the relative difference between the transmission and reception sides.

4. This value as an example is calculated when the conditions described in the "Conditions" column are met. Refer to Note 3 above to calculate the maximum transfer rate under conditions of the customer.

**Caution** Select the TTL input buffer for RxDq and the N-ch open-drain output (V<sub>DD</sub> tolerance) mode for TxDq by using the PIMg and POMg registers.

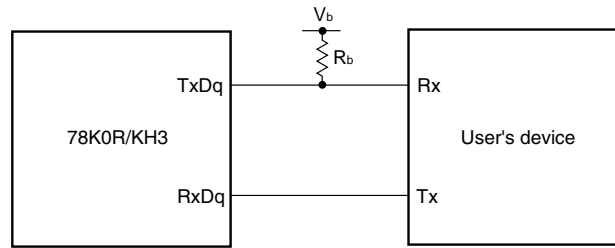
(Remarks are given on the next page.)

**(3) Serial interface: Serial array unit (9/18)**

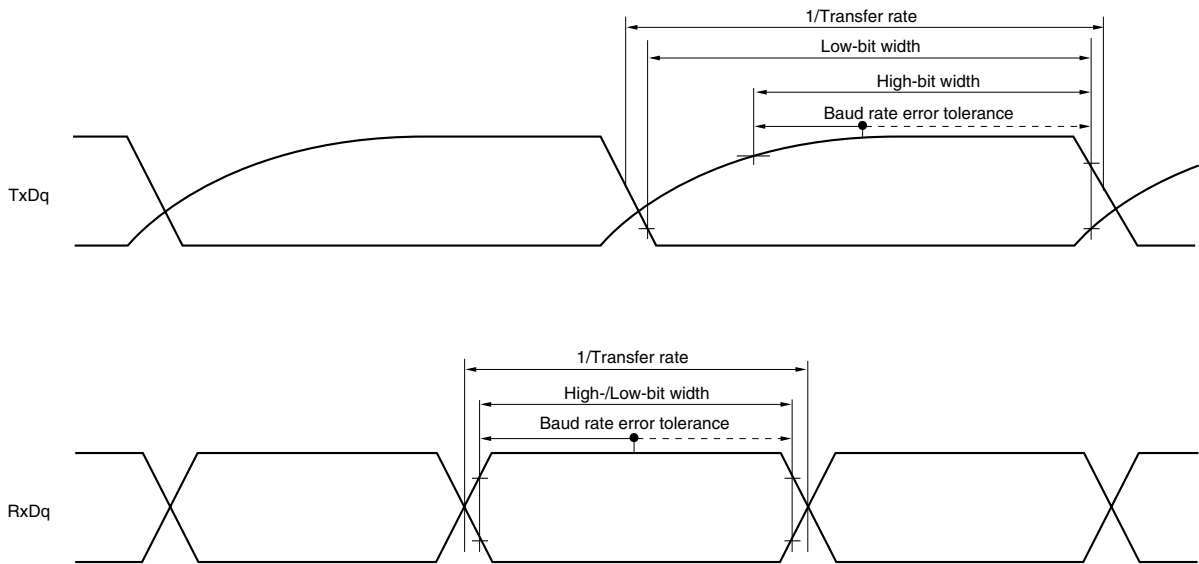
- Remarks**
1.  $R_b[\Omega]$ : Communication line (TxDq) pull-up resistance,  
 $C_b[F]$ : Communication line (TxDq) load capacitance,  $V_b[V]$ : Communication line voltage
  2. q: UART number (q = 1, 2), g: PIM and POM number (g = 0, 14)
  3.  $f_{MCK}$ : Serial array unit operation clock frequency  
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1),  
n: Channel number (n = 0 to 3))
  4.  $V_{IH}$  and  $V_{IL}$  below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.  
 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}, 2.7\text{ V} \leq V_b \leq 4.0\text{ V}: V_{IH} = 2.2\text{ V}, V_{IL} = 0.8\text{ V}$   
 $2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}, 2.3\text{ V} \leq V_b \leq 2.7\text{ V}: V_{IH} = 2.0\text{ V}, V_{IL} = 0.5\text{ V}$
  5. UART0 and UART3 cannot communicate at different potential. Use UART1 and UART2 for communication at different potential.

## (3) Serial interface: Serial array unit (10/18)

## UART mode connection diagram (during communication at different potential)



## UART mode bit width (during communication at different potential) (reference)



**Caution** Select the TTL input buffer for RxDq and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for TxDq by using the PIMg and POMg registers.

- Remarks**
1.  $R_b[\Omega]$ : Communication line (TxDq) pull-up resistance,  $V_b[V]$ : Communication line voltage
  2. q: UART number (q = 1, 2), g: PIM and POM number (g = 0, 14)
  3. UART0 and UART3 cannot communicate at different potential. Use UART1 and UART2 for communication at different potential.

## (3) Serial interface: Serial array unit (11/18)

(TA = -40 to +85°C, 2.7 V ≤ VDD = EVDD0 = EVDD1 ≤ 5.5 V, VSS = EVSS0 = EVSS1 = AVSS = 0 V)

<R> (f) During communication at different potential (2.5 V, 3 V) (CSI mode) (master mode,  $\overline{\text{SCKp}}$ ... internal clock output) (1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t <sub>KCY1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	400 <sup>Note 1</sup>			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	800 <sup>Note 1</sup>			ns
$\overline{\text{SCKp}}$ high-level width	t <sub>KH1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	t <sub>KCY1</sub> /2 - 75			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	t <sub>KCY1</sub> /2 - 170			ns
$\overline{\text{SCKp}}$ low-level width	t <sub>KL1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	t <sub>KCY1</sub> /2 - 20			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	t <sub>KCY1</sub> /2 - 35			ns
Slp setup time (to $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 2</sup>	t <sub>SIK1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	150			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	275			ns
Slp hold time (from $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 2</sup>	t <sub>KSI1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	30			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	30			ns
Delay time from $\overline{\text{SCKp}}\downarrow$ to SO <sub>p</sub> output <sup>Note 2</sup>	t <sub>KSO1</sub>	4.0 V ≤ V <sub>DD</sub> < 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ			120	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ			215	ns

**Notes 1.** The value must also be 4/f<sub>CLK</sub> or more.2. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1.**Caution** Select the TTL input buffer for Slp and the N-ch open-drain output (V<sub>DD</sub> tolerance) mode for SO<sub>p</sub> and  $\overline{\text{SCKp}}$  by using the PIMg and POMg registers.**Remarks 1.** p: CSI number (p = 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

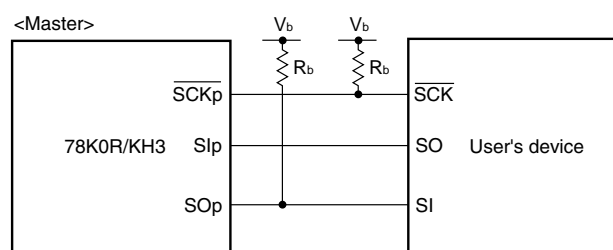
3. R<sub>b</sub>[Ω]: Communication line ( $\overline{\text{SCKp}}$ , SO<sub>p</sub>) pull-up resistance,C<sub>b</sub>[F]: Communication line ( $\overline{\text{SCKp}}$ , SO<sub>p</sub>) load capacitance, V<sub>b</sub>[V]: Communication line voltage4. V<sub>IH</sub> and V<sub>IL</sub> below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.4.0 V ≤ V<sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V<sub>b</sub> ≤ 4.0 V: V<sub>IH</sub> = 2.2 V, V<sub>IL</sub> = 0.8 V2.7 V ≤ V<sub>DD</sub> < 4.0 V, 2.3 V ≤ V<sub>b</sub> ≤ 2.7 V: V<sub>IH</sub> = 2.0 V, V<sub>IL</sub> = 0.5 V

5. CSI00 cannot communicate at different potential. Use CSI01, CSI10, CSI11, CSI20, and CSI21 for communication at different potential.

## (3) Serial interface: Serial array unit (12/18)

(T<sub>A</sub> = -40 to +85°C, 2.7 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)<R> (f) During communication at different potential (2.5 V, 3 V) (CSI mode) (master mode,  $\overline{\text{SCKp}}$ ... internal clock output) (2/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Slp setup time (to $\overline{\text{SCKp}}$ ↓) <sup>Note</sup>	t <sub>SIK1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	70			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	100			ns
Slp hold time (from $\overline{\text{SCKp}}$ ↓) <sup>Note</sup>	t <sub>KS11</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	30			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	30			ns
Delay time from $\overline{\text{SCKp}}$ ↑ to SO <sub>p</sub> output <sup>Note</sup>	t <sub>KS01</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ			40	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ			40	ns

**Note** When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.**CSI mode connection diagram (during communication at different potential)****Caution** Select the TTL input buffer for Slp and the N-ch open-drain output (V<sub>DD</sub> tolerance) mode for SOp and  $\overline{\text{SCKp}}$  by using the PIMg and POMg registers.**Remarks 1.** p: CSI number (p = 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14)

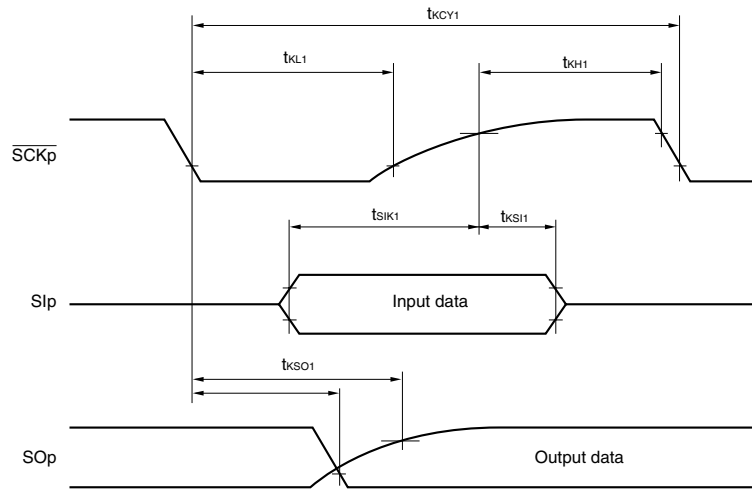
2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

3. R<sub>b</sub>[Ω]: Communication line ( $\overline{\text{SCKp}}$ , SO<sub>p</sub>) pull-up resistance,C<sub>b</sub>[F]: Communication line ( $\overline{\text{SCKp}}$ , SO<sub>p</sub>) load capacitance, V<sub>b</sub>[V]: Communication line voltage4. V<sub>IH</sub> and V<sub>IL</sub> below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.4.0 V ≤ V<sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V<sub>b</sub> ≤ 4.0 V: V<sub>IH</sub> = 2.2 V, V<sub>IL</sub> = 0.8 V2.7 V ≤ V<sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V<sub>b</sub> ≤ 2.7 V: V<sub>IH</sub> = 2.0 V, V<sub>IL</sub> = 0.5 V

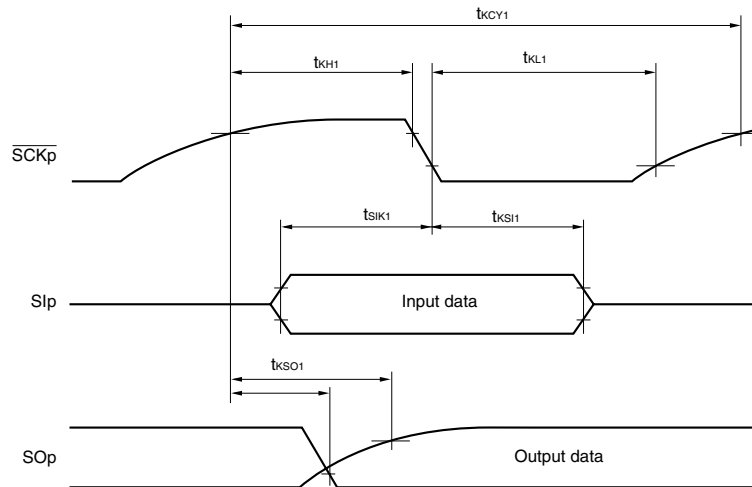
5. CSI00 cannot communicate at different potential. Use CSI01, CSI10, CSI11, CSI20, and CSI21 for communication at different potential.

## (3) Serial interface: Serial array unit (13/18)

**CSI mode serial transfer timing (during communication at different potential)**  
**(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)**



**CSI mode serial transfer timing (during communication at different potential)**  
**(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)**



**Caution** Select the TTL input buffer for  $Slp$  and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for  $SOp$  and  $\overline{SCKp}$  by using the PIMg and POMg registers.

- Remarks**
1. p: CSI number (p = 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14)
  2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
  3. CSI00 cannot communicate at different potential. Use CSI01, CSI10, CSI11, CSI20, and CSI21 for communication at different potential.



(3) Serial interface: Serial array unit (14/18)

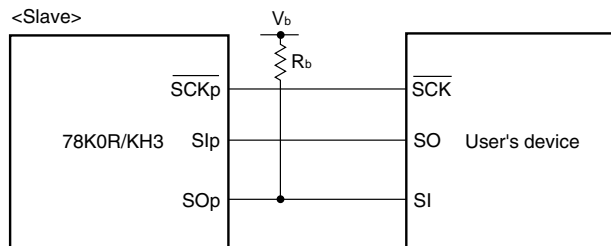
(T<sub>A</sub> = -40 to +85°C, 2.7 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)

<R> (g) During communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode,  $\overline{\text{SCKp}}$ ... external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t <sub>KCY2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V	13.6 MHz < f <sub>MCK</sub>	10/f <sub>MCK</sub>		ns
			6.8 MHz < f <sub>MCK</sub> ≤ 13.6 MHz	8/f <sub>MCK</sub>		ns
			f <sub>MCK</sub> ≤ 6.8 MHz	6/f <sub>MCK</sub>		ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V	18.5 MHz < f <sub>MCK</sub>	16/f <sub>MCK</sub>		ns
			14.8 MHz < f <sub>MCK</sub> ≤ 18.5 MHz	14/f <sub>MCK</sub>		ns
			11.1 MHz < f <sub>MCK</sub> ≤ 14.8 MHz	12/f <sub>MCK</sub>		ns
			7.4 MHz < f <sub>MCK</sub> ≤ 11.1 MHz	10/f <sub>MCK</sub>		ns
			3.7 MHz < f <sub>MCK</sub> ≤ 7.4 MHz	8/f <sub>MCK</sub>		ns
f <sub>MCK</sub> ≤ 3.7 MHz	6/f <sub>MCK</sub>		ns			
$\overline{\text{SCKp}}$ high-/low-level width	t <sub>KH2</sub> ,	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V	f <sub>KCY2</sub> /2 - 20			ns
	t <sub>KL2</sub>	2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V	f <sub>KCY2</sub> /2 - 35			ns
Slp setup time (to $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 1</sup>	t <sub>SIK2</sub>		90			ns
Slp hold time (from $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 2</sup>	t <sub>SI2</sub>		1/f <sub>MCK</sub> + 50			ns
Delay time from $\overline{\text{SCKp}}\downarrow$ to SOp output <sup>Note 3</sup>	t <sub>KSO2</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ			2/f <sub>MCK</sub> + 120	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ			2/f <sub>MCK</sub> + 230	ns

- Notes**
- When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The Slp setup time becomes “to  $\overline{\text{SCKp}}\downarrow$ ” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
  - When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The Slp hold time becomes “from  $\overline{\text{SCKp}}\downarrow$ ” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.
  - When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1. The delay time to SOp output becomes “from  $\overline{\text{SCKp}}\uparrow$ ” when DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.

CSI mode connection diagram (during communication at different potential)



(Caution and Remarks are given on the next page.)

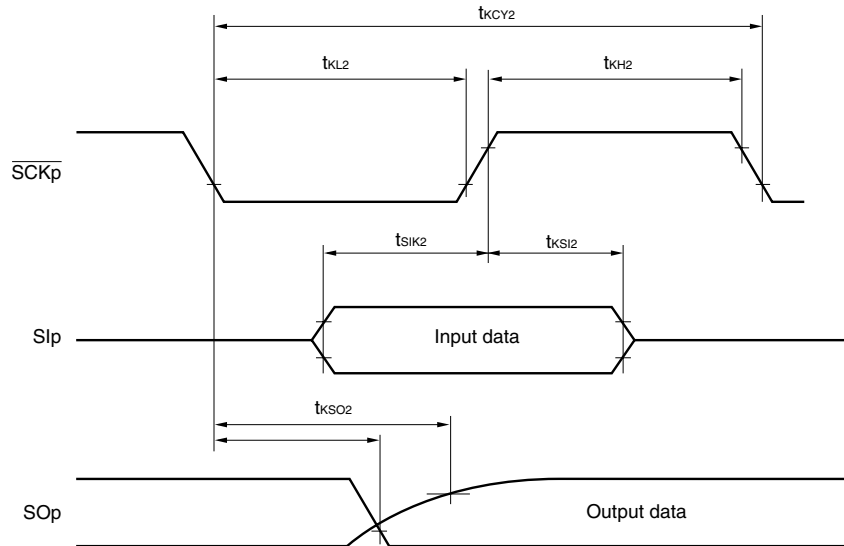
**(3) Serial interface: Serial array unit (15/18)**

**Caution** Select the TTL input buffer for  $S_{Ip}$  and  $\overline{SCKp}$  and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for  $S_{Op}$  by using the  $PIMg$  and  $POMg$  registers.

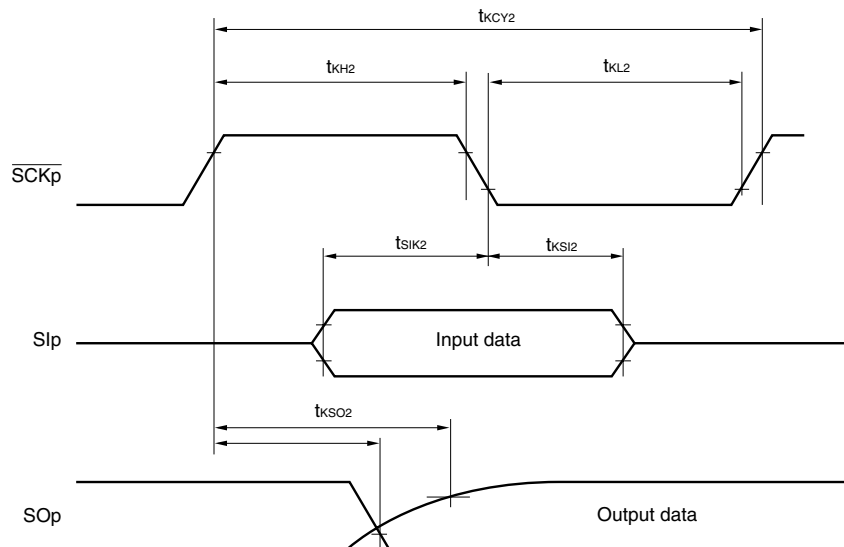
- Remarks**
1. p: CSI number (p = 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14)
  2.  $R_b[\Omega]$ : Communication line (SOp) pull-up resistance,  
 $C_b[F]$ : Communication line (SOp) load capacitance,  $V_b[V]$ : Communication line voltage
  3.  $f_{MCK}$ : Serial array unit operation clock frequency  
 (Operation clock to be set by the  $CKSmn$  bit of the  $SMRmn$  register. m: Unit number (m = 0, 3),  
 n: Channel number (n = 0 to 2))
  4.  $V_{IH}$  and  $V_{IL}$  below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.  
 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}, 2.7\text{ V} \leq V_b \leq 4.0\text{ V}: V_{IH} = 2.2\text{ V}, V_{IL} = 0.8\text{ V}$   
 $2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}, 2.3\text{ V} \leq V_b \leq 2.7\text{ V}: V_{IH} = 2.0\text{ V}, V_{IL} = 0.5\text{ V}$
  5. CSI00 cannot communicate at different potential. Use CSI01, CSI10, CSI11, CSI20, and CSI21 for communication at different potential.

## (3) Serial interface: Serial array unit (16/18)

CSI mode serial transfer timing (during communication at different potential)  
 (When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



CSI mode serial transfer timing (during communication at different potential)  
 (When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



**Caution** Select the TTL input buffer for  $Slp$  and  $\overline{SCKp}$  and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for  $SOp$  by using the PIMg and POMg registers.

- Remarks**
1. p: CSI number (p = 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14)
  2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
  3. CSI00 cannot communicate at different potential. Use CSI01, CSI10, CSI11, CSI20, and CSI21 for communication at different potential.

## (3) Serial interface: Serial array unit (17/18)

(T<sub>A</sub> = -40 to +85°C, 2.7 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)(h) During communication at different potential (2.5 V, 3 V) (simplified I<sup>2</sup>C mode)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
<R> SCLr clock frequency	f <sub>SCL</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 1.4 kΩ		400 <sup>Note</sup>	kHz
		2.7 V ≤ V <sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ		400 <sup>Note</sup>	kHz
Hold time when SCLr = "L"	t <sub>LOW</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 1.4 kΩ	1065		ns
		2.7 V ≤ V <sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	1065		ns
Hold time when SCLr = "H"	t <sub>HIGH</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 1.4 kΩ	445		ns
		2.7 V ≤ V <sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	445		ns
Data setup time (reception)	t <sub>SU:DAT</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 1.4 kΩ	1/f <sub>MCK</sub> + 190		ns
		2.7 V ≤ V <sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	1/f <sub>MCK</sub> + 190		ns
Data hold time (transmission)	t <sub>HD:DAT</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 1.4 kΩ	0	160	ns
		2.7 V ≤ V <sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	0	160	ns

<R> **Note** The value must also be f<sub>MCK</sub>/4 or less.

**Caution** Select the TTL input buffer and the N-ch open-drain output (V<sub>DD</sub> tolerance) mode for SDAr and the N-ch open-drain output (V<sub>DD</sub> tolerance) mode for SCLr by using the PIMg and POMg registers.

**Remarks 1.** R<sub>b</sub>[Ω]: Communication line (SDAr, SCLr) pull-up resistance,

C<sub>b</sub>[F]: Communication line (SDAr, SCLr) load capacitance, V<sub>b</sub>[V]: Communication line voltage

**2.** r: IIC number (r = 10, 11, 20, 21), g: PIM, POM number (g = 0, 9, 12, 14)

**3.** f<sub>MCK</sub>: Serial array unit operation clock frequency

(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1),

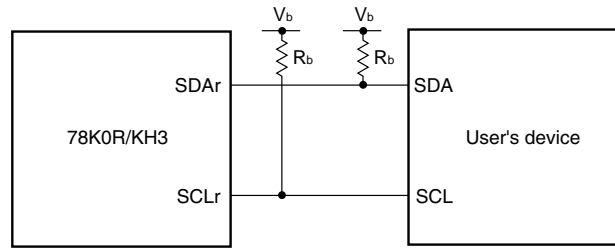
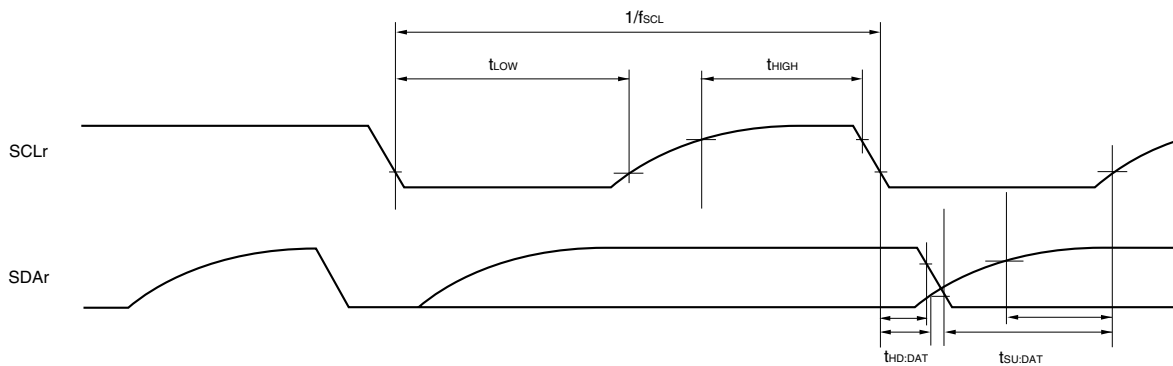
n: Channel number (n = 0, 3), mn = 02, 03, 10, 11)

**4.** V<sub>IH</sub> and V<sub>IL</sub> below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in simplified I<sup>2</sup>C mode.

4.0 V ≤ V<sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V<sub>b</sub> ≤ 4.0 V: V<sub>IH</sub> = 2.2 V, V<sub>IL</sub> = 0.8 V

2.7 V ≤ V<sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V<sub>b</sub> ≤ 2.7 V: V<sub>IH</sub> = 2.0 V, V<sub>IL</sub> = 0.5 V

## (3) Serial interface: Serial array unit (18/18)

Simplified I<sup>2</sup>C mode connection diagram (during communication at different potential)Simplified I<sup>2</sup>C mode serial transfer timing (during communication at different potential)

**Caution** Select the TTL input buffer and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for SDAr and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for SCLr by using the PIMg and POMg registers.

**Remarks 1.**  $R_b[\Omega]$ : Communication line (SDAr, SCLr) pull-up resistance,  $V_b[V]$ : Communication line voltage  
**2.** r: IIC number ( $r = 10, 11, 20, 21$ ), g: PIM and POM number ( $g = 0, 9, 12, 14$ )

## (4) Serial interface: IIC0

 $(T_A = -40 \text{ to } +85^\circ\text{C}, 1.8 \text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5 \text{ V}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 \text{ V})$ 

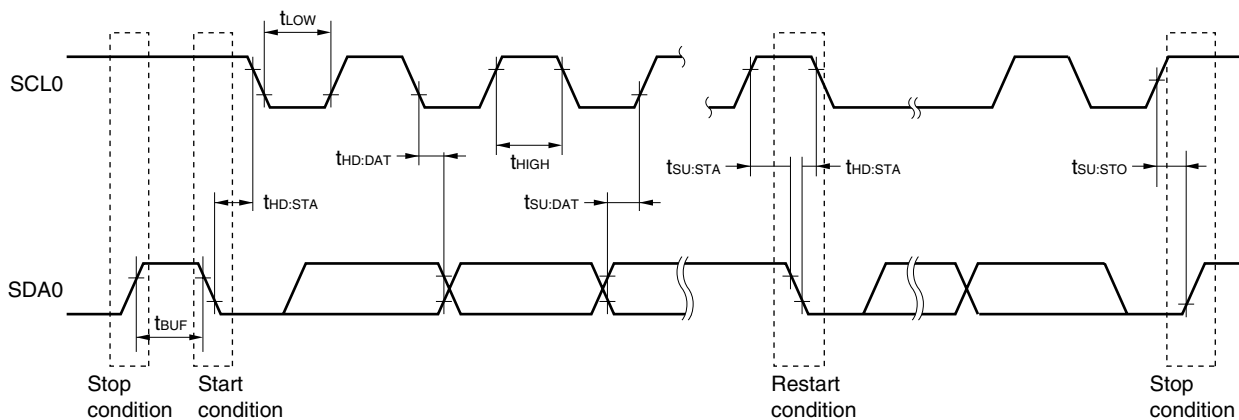
## (a) IIC0

Parameter	Symbol	Conditions	Standard Mode		Fast Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCL0 clock frequency	$f_{SCL}$	$6.7 \text{ MHz} \leq f_{CLK}$	0	100	0	400	kHz
		$4.0 \text{ MHz} \leq f_{CLK} < 6.7 \text{ MHz}$	0	100	0	340	kHz
		$3.2 \text{ MHz} \leq f_{CLK} < 4.0 \text{ MHz}$	0	100	–	–	kHz
		$2.0 \text{ MHz} \leq f_{CLK} < 3.2 \text{ MHz}$	0	85	–	–	kHz
Setup time of restart condition <sup>Note 1</sup>	$t_{SU:STA}$		4.7		0.6		$\mu\text{s}$
Hold time	$t_{HD:STA}$		4.0		0.6		$\mu\text{s}$
Hold time when SCL0 = "L"	$t_{LOW}$		4.7		1.3		$\mu\text{s}$
Hold time when SCL0 = "H"	$t_{HIGH}$		4.0		0.6		$\mu\text{s}$
Data setup time (reception)	$t_{SU:DAT}$		250		100		ns
Data hold time (transmission) <sup>Note 2</sup>	$t_{HD:DAT}$	$CL00 = 1 \text{ and } CL01 = 1$	0	3.45 <sup>Note 3</sup>	0	0.9 <sup>Note 4</sup>	$\mu\text{s}$
				5.50 <sup>Note 5</sup>		1.5 <sup>Note 6</sup>	$\mu\text{s}$
		$CL00 = 0 \text{ and } CL01 = 0, \text{ or } CL00 = 1 \text{ and } CL01 = 0$	0	3.45	0	0.9 <sup>Note 7</sup>	$\mu\text{s}$
						0.95 <sup>Note 8</sup>	$\mu\text{s}$
$CL00 = 0 \text{ and } CL01 = 1$	0	3.45	0	0.9	$\mu\text{s}$		
Setup time of stop condition	$t_{SU:STO}$		4.0		0.6		$\mu\text{s}$
Bus-free time	$t_{BUF}$		4.7		1.3		$\mu\text{s}$

- Notes**
- The first clock pulse is generated after this period when the start/restart condition is detected.
  - The maximum value (MAX.) of  $t_{HD:DAT}$  is during normal transfer and a wait state is inserted in the  $\overline{ACK}$  (acknowledge) timing.
  - When  $3.2 \text{ MHz} \leq f_{CLK} \leq 4.19 \text{ MHz}$ .
  - When  $6.7 \text{ MHz} \leq f_{CLK} \leq 8.38 \text{ MHz}$ .
  - When  $2.0 \text{ MHz} \leq f_{CLK} < 3.2 \text{ MHz}$ . At this time, use the SCL0 clock within 85 kHz.
  - When  $4.0 \text{ MHz} \leq f_{CLK} < 6.7 \text{ MHz}$ . At this time, use the SCL0 clock within 340 kHz.
  - When  $8.0 \text{ MHz} \leq f_{CLK} \leq 16.76 \text{ MHz}$ .
  - When  $7.6 \text{ MHz} \leq f_{CLK} < 8.0 \text{ MHz}$ .

**Remark** CL00, CL01, DFC0: Bits 0, 1, and 2 of the IIC clock select register 0 (IICCL0)

## IIC0 serial transfer timing



**(5) Serial interface: On-chip debug (UART)****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )****(a) On-chip debug (UART)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate			$f_{CLK}/2^{12}$		$f_{CLK}/6$	bps
		Flash memory programming mode			2.66	Mbps
TOOL1 output frequency	$f_{TOOL1}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			10	MHz
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			2.5	MHz

**A/D Converter Characteristics (1/2)**(T<sub>A</sub> = -40 to +85°C, 2.3 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, 2.3 V ≤ AV<sub>REF0</sub> ≤ V<sub>DD</sub>, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)**(a) Conventional-specification products (μPD78F117x)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	R <sub>ES</sub>				10	bit
Overall error <sup>Notes 1, 2</sup>	AINL	4.0 V ≤ AV <sub>REF0</sub> ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV <sub>REF0</sub> < 4.0 V			±0.6	%FSR
		2.3 V ≤ AV <sub>REF0</sub> < 2.7 V			±0.7	%FSR
Conversion time	t <sub>CONV</sub>	4.0 V ≤ AV <sub>REF0</sub> ≤ 5.5 V	6.1		66.6	μs
		2.7 V ≤ AV <sub>REF0</sub> < 4.0 V	12.2		66.6	μs
		2.3 V ≤ AV <sub>REF0</sub> < 2.7 V	27		66.6	μs
Zero-scale error <sup>Notes 1, 2</sup>	E <sub>ZS</sub>	4.0 V ≤ AV <sub>REF0</sub> ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV <sub>REF0</sub> < 4.0 V			±0.6	%FSR
		2.3 V ≤ AV <sub>REF0</sub> < 2.7 V			±0.6	%FSR
Full-scale error <sup>Notes 1, 2</sup>	E <sub>FS</sub>	4.0 V ≤ AV <sub>REF0</sub> ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV <sub>REF0</sub> < 4.0 V			±0.6	%FSR
		2.3 V ≤ AV <sub>REF0</sub> < 2.7 V			±0.6	%FSR
Integral linearity error <sup>Note 1</sup>	I <sub>LE</sub>	4.0 V ≤ AV <sub>REF0</sub> ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV <sub>REF0</sub> < 4.0 V			±4.5	LSB
		2.3 V ≤ AV <sub>REF0</sub> < 2.7 V			±4.5	LSB
Differential linearity error <sup>Note 1</sup>	D <sub>LE</sub>	4.0 V ≤ AV <sub>REF0</sub> ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV <sub>REF0</sub> < 4.0 V			±2.0	LSB
		2.3 V ≤ AV <sub>REF0</sub> < 2.7 V			±2.0	LSB
Analog input voltage	V <sub>AIN</sub>	2.3 V ≤ AV <sub>REF0</sub> ≤ 5.5 V	AV <sub>SS</sub>		AV <sub>REF0</sub>	V

**Notes** 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.



**A/D Converter Characteristics (2/2)**(T<sub>A</sub> = -40 to +85°C, 2.3 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, 2.3 V ≤ AV<sub>REF0</sub> ≤ V<sub>DD</sub>, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)**(b) Expanded-specification products (μPD78F117xA)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	R <sub>ES</sub>				10	bit
Overall error <sup>Notes 1, 2</sup>	AINL	4.0 V ≤ AV <sub>REF0</sub> ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV <sub>REF0</sub> < 4.0 V			±0.5	%FSR
		2.3 V ≤ AV <sub>REF0</sub> < 2.7 V			±0.7	%FSR
Conversion time	t <sub>CONV</sub>	4.0 V ≤ AV <sub>REF0</sub> ≤ 5.5 V	6.1		66.6	μs
		2.7 V ≤ AV <sub>REF0</sub> < 4.0 V	12.2		66.6	μs
		2.3 V ≤ AV <sub>REF0</sub> < 2.7 V	27		66.6	μs
Zero-scale error <sup>Notes 1, 2</sup>	E <sub>ZS</sub>	4.0 V ≤ AV <sub>REF0</sub> ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV <sub>REF0</sub> < 4.0 V			±0.5	%FSR
		2.3 V ≤ AV <sub>REF0</sub> < 2.7 V			±0.5	%FSR
Full-scale error <sup>Notes 1, 2</sup>	E <sub>FS</sub>	4.0 V ≤ AV <sub>REF0</sub> ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV <sub>REF0</sub> < 4.0 V			±0.5	%FSR
		2.3 V ≤ AV <sub>REF0</sub> < 2.7 V			±0.5	%FSR
Integral linearity error <sup>Note 1</sup>	I <sub>LE</sub>	4.0 V ≤ AV <sub>REF0</sub> ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV <sub>REF0</sub> < 4.0 V			±3.5	LSB
		2.3 V ≤ AV <sub>REF0</sub> < 2.7 V			±3.5	LSB
Differential linearity error <sup>Note 1</sup>	D <sub>LE</sub>	4.0 V ≤ AV <sub>REF0</sub> ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV <sub>REF0</sub> < 4.0 V			±1.5	LSB
		2.3 V ≤ AV <sub>REF0</sub> < 2.7 V			±1.5	LSB
Analog input voltage	V <sub>AIN</sub>	2.3 V ≤ AV <sub>REF0</sub> ≤ 5.5 V	AV <sub>SS</sub>		AV <sub>REF0</sub>	V

**Notes** 1. Excludes quantization error (±1/2 LSB).

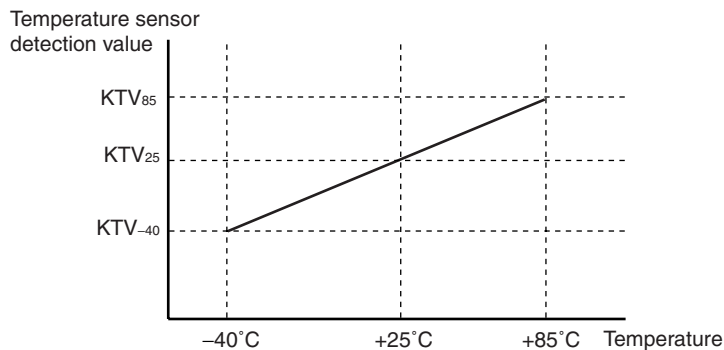
2. This value is indicated as a ratio (%FSR) to the full-scale value.

**Temperature Sensor (Expanded-Specification Products ( $\mu$ PD78F117xA) Only)****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Augmentation factor per $10^\circ\text{C}$	TC		1	3.5	15	$/10^\circ\text{C}$
Temperature sensor detection value	KTV <sub>-40</sub>	$T_A = -40^\circ\text{C}$	30	80	130	–
	KTV <sub>25</sub>	$T_A = +25^\circ\text{C}$	65	101	140	–
	KTV <sub>85</sub>	$T_A = +85^\circ\text{C}$	100	122	150	–

**Remark** The temperature sensor detection value is obtained by using the following expression.

$$\text{Temperature sensor detection value} = \frac{\text{A/D conversion value with sensor that depends on temperature}}{\text{A/D conversion value with sensor that does not depend on temperature}} \times 256 \cong \frac{TC}{10} \left( \text{Temperature during sensor operation} - \text{Low reference temperature} \right) + \text{Temperature sensor detection value at a low reference temperature}$$

**D/A Converter Characteristics****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )**

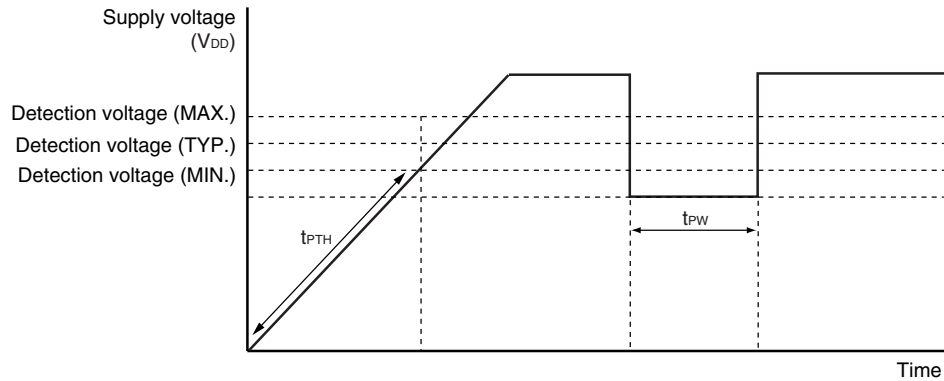
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Resolution	RES				8	bit	
Overall error	AINL	$R_{load} = 2\text{ M}\Omega$			$\pm 1.2$	%FSR	
		$R_{load} = 4\text{ M}\Omega$			$\pm 0.8$	%FSR	
		$R_{load} = 10\text{ M}\Omega$			$\pm 0.6$	%FSR	
Settling time	$t_{SET}$	$C_{load} = 20\text{ pF}$	$4.0\text{ V} \leq AV_{REF1} \leq 5.5\text{ V}$			3	$\mu\text{s}$
			$2.7\text{ V} \leq AV_{REF1} < 4.0\text{ V}$			3	$\mu\text{s}$
			$1.8\text{ V} \leq AV_{REF1} < 2.7\text{ V}$			6	$\mu\text{s}$
D/A output resistance value	$R_o$	Per D/A converter channel		6.4		$\text{k}\Omega$	

**Remark** When the D/A converter is in normal mode, D/A conversion is started after one  $f_{CLK}$  clock has elapsed since the DACSn register was written. The output level is determined when the settling time has elapsed after the D/A conversion was started.

**POC Circuit Characteristics ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	$V_{POC0}$		1.5	1.59	1.68	V
Power supply voltage rise inclination	$t_{PTH}$	Change inclination of $V_{DD}$ : $0\text{ V} \rightarrow V_{POC0}$	0.5			V/ms
Minimum pulse width	$t_{PW}$	When the voltage drops	200			$\mu\text{s}$
Detection delay time					200	$\mu\text{s}$

**POC Circuit Timing**



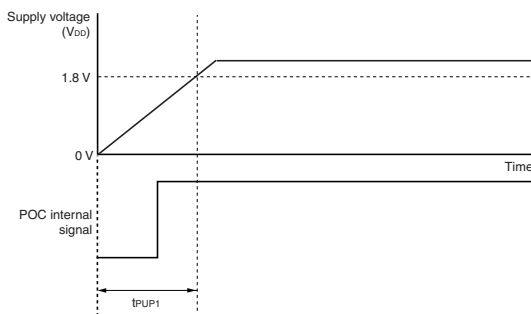
**Supply Voltage Rise Time ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 1.8 V ( $V_{DD}(\text{MIN.})$ ) <sup>Note</sup> ( $V_{DD}$ : $0\text{ V} \rightarrow 1.8\text{ V}$ )	$t_{PUP1}$	LVI default start function stopped is set ( $\overline{\text{LVIOFF}}$ (option byte) = 1), when $\overline{\text{RESET}}$ input is not used			3.6	ms
Maximum time to rise to 1.8 V ( $V_{DD}(\text{MIN.})$ ) <sup>Note</sup> (releasing $\overline{\text{RESET}}$ input $\rightarrow V_{DD}$ : 1.8 V)	$t_{PUP2}$	LVI default start function stopped is set ( $\overline{\text{LVIOFF}}$ (option byte) = 1), when $\overline{\text{RESET}}$ input is used			1.88	ms

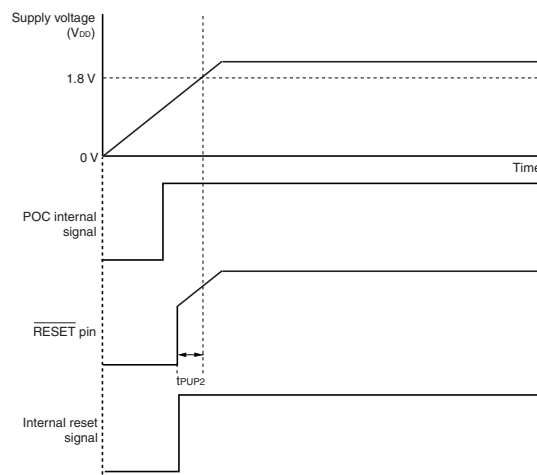
**Note** Make sure to raise the power supply in a shorter time than this.

**Supply Voltage Rise Time Timing**

- When  $\overline{\text{RESET}}$  pin input is not used



- When  $\overline{\text{RESET}}$  pin input is used (when external reset is released by the  $\overline{\text{RESET}}$  pin, after POC has been released)



LVI Circuit Characteristics ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{POC} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5$  V,  $V_{SS} = EV_{SS0} = EV_{SS1} = 0$  V)

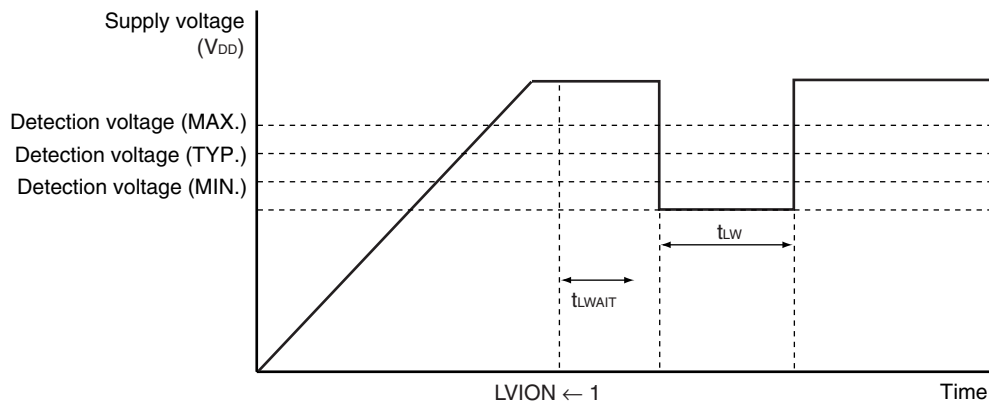
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Detection voltage	Supply voltage level	$V_{LVI0}$		4.12	4.22	4.32	V
		$V_{LVI1}$		3.97	4.07	4.17	V
		$V_{LVI2}$		3.82	3.92	4.02	V
		$V_{LVI3}$		3.66	3.76	3.86	V
		$V_{LVI4}$		3.51	3.61	3.71	V
		$V_{LVI5}$		3.35	3.45	3.55	V
		$V_{LVI6}$		3.20	3.30	3.40	V
		$V_{LVI7}$		3.05	3.15	3.25	V
		$V_{LVI8}$		2.89	2.99	3.09	V
		$V_{LVI9}$		2.74	2.84	2.94	V
		$V_{LVI10}$		2.58	2.68	2.78	V
		$V_{LVI11}$		2.43	2.53	2.63	V
		$V_{LVI12}$		2.28	2.38	2.48	V
		$V_{LVI13}$		2.12	2.22	2.32	V
		$V_{LVI14}$		1.97	2.07	2.17	V
$V_{LVI15}$		1.81	1.91	2.01	V		
External input pin <sup>Note 1</sup>	$V_{EXLVI}$	$EXLVI < V_{DD}$ , $1.8 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	1.11	1.21	1.31	V	
Power supply voltage on power application	$V_{PULVI}$	When LVI default start function enabled is set	1.87	2.07	2.27	V	
Minimum pulse width	$t_{LW}$		200			$\mu\text{s}$	
Detection delay time					200	$\mu\text{s}$	
Operation stabilization wait time <sup>Note 2</sup>	$t_{LWAIT}$				10	$\mu\text{s}$	

**Notes** 1. The EXLVI/P120/INTP0 pin is used.

2. Time required from setting bit 7 (LVION) of the low-voltage detection register (LVIM) to 1 to operation stabilization

**Remark**  $V_{LVI(n-1)} > V_{LVI n}$ :  $n = 1$  to  $15$

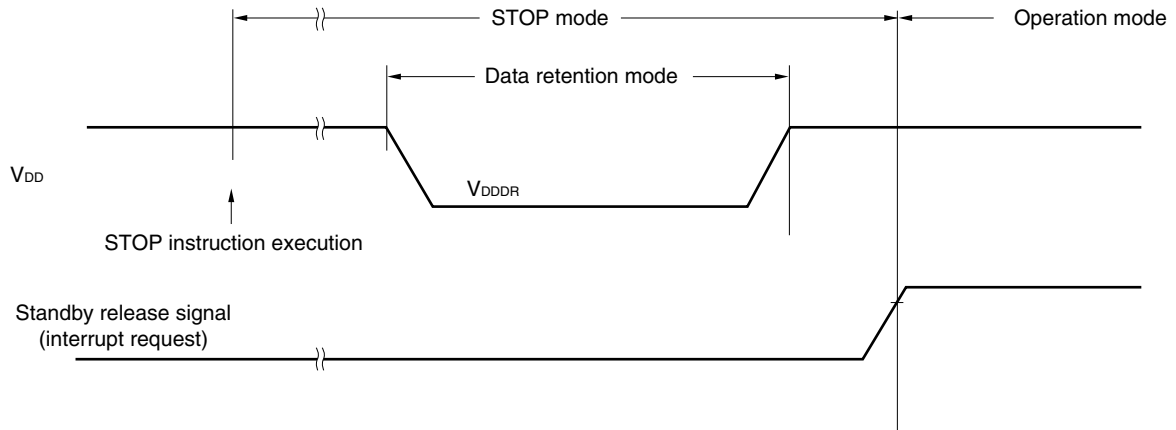
## LVI Circuit Timing



**Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics ( $T_A = -40$  to  $+85^\circ\text{C}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	$V_{DDDR}$		1.5 <sup>Note</sup>		5.5	V

**Note** The value depends on the POC detection voltage. When the voltage drops, the data is retained until a POC reset is effected, but data is not retained when a POC reset is effected.



**Flash Memory Programming Characteristics**(T<sub>A</sub> = -40 to +85°C, 2.7 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = 0 V)**(a) Conventional-specification products (μPD78F117x)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
V <sub>DD</sub> supply current	I <sub>DD</sub>	TYP. = 10 MHz, MAX. = 20 MHz			4.5	15	mA
CPU/peripheral hardware clock frequency	f <sub>CLK</sub>			2		20	MHz
Number of rewrites (number of deletes per block)	C <sub>WRT</sub>	Used for updating programs When using flash memory programmer and NEC Electronics self programming library	Retained for 15 years	100			Times
		Used for updating data When using NEC Electronics EEPROM emulation library (usable ROM size: 6 KB of 3 consecutive blocks)	Retained for 3 years	10,000			Times

**Remark** When updating data multiple times, use the flash memory as one for updating data.**(b) Expanded-specification products (μPD78F117xA)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
V <sub>DD</sub> supply current	I <sub>DD</sub>	TYP. = 10 MHz, MAX. = 20 MHz			4.5	15	mA
CPU/peripheral hardware clock frequency	f <sub>CLK</sub>			2		20	MHz
<R> Number of rewrites (number of deletes per block)	C <sub>WRT</sub>	Used for updating programs When using flash memory programmer and NEC Electronics self programming library	Retained for 15 years	1000			Times
		Used for updating data When using NEC Electronics EEPROM emulation library (usable ROM size: 6 KB of 3 consecutive blocks)	Retained for 5 years	10,000			Times

**Remark** When updating data multiple times, use the flash memory as one for updating data.

## CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)

Target products  $\mu$ PD78F1174A(A), 78F1175A(A), 78F1176A(A), 78F1177A(A), 78F1178A(A)

**Cautions** The 78K0R/KH3 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. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.

### Absolute Maximum Ratings (T<sub>A</sub> = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V <sub>DD</sub>		-0.5 to +6.5	V
	EV <sub>DD0</sub> , EV <sub>DD1</sub>	EV <sub>DD0</sub> = EV <sub>DD1</sub>	-0.5 to +6.5	V
	V <sub>SS</sub>		-0.5 to +0.3	V
	EV <sub>SS0</sub> , EV <sub>SS1</sub>	EV <sub>SS0</sub> = EV <sub>SS1</sub>	-0.5 to +0.3	V
	AV <sub>REF0</sub>		-0.5 to V <sub>DD</sub> +0.3 <sup>Note 1</sup>	V
	AV <sub>REF1</sub>		-0.5 to V <sub>DD</sub> +0.3 <sup>Note 1</sup>	V
	AV <sub>SS</sub>		-0.5 to +0.3	V
REGC pin input voltage	V <sub>REGC</sub>	REGC	-0.3 to +3.6 and -0.3 to V <sub>DD</sub> +0.3 <sup>Note 2</sup>	V
Input voltage	V <sub>I1</sub>	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120 to P127, P131, P140 to P145, P160 to P163, EXCLK, RESET, FLMD0	-0.3 to EV <sub>DD0</sub> , EV <sub>DD1</sub> +0.3 and -0.3 to V <sub>DD</sub> +0.3 <sup>Note 1</sup>	V
	V <sub>I2</sub>	P60 to P63 (N-ch open-drain)	-0.3 to +6.5	V
	V <sub>I3</sub>	P20 to P27, P150 to P157	-0.3 to AV <sub>REF0</sub> +0.3 and -0.3 to V <sub>DD</sub> +0.3 <sup>Note 1</sup>	V
	V <sub>I4</sub>	P110, P111	-0.3 to AV <sub>REF1</sub> +0.3 and -0.3 to V <sub>DD</sub> +0.3 <sup>Note 1</sup>	V
Output voltage	V <sub>O1</sub>	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	-0.3 to EV <sub>DD0</sub> , EV <sub>DD1</sub> +0.3 <sup>Note 1</sup>	V
	V <sub>O2</sub>	P20 to P27, P150 to P157	-0.3 to AV <sub>REF0</sub> +0.3	V
	V <sub>O3</sub>	P110, P111	-0.3 to AV <sub>REF1</sub> +0.3	V

**Notes 1.** Must be 6.5 V or lower.

- 2.** Connect the REGC pin to V<sub>SS</sub> via a capacitor (0.47 to 1  $\mu$ F). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

**Caution** Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**Absolute Maximum Ratings (T<sub>A</sub> = 25°C) (2/2)**

Parameter	Symbols	Conditions		Ratings	Unit
Analog input voltage	V <sub>AN</sub>	ANI0 to ANI15		-0.3 to AV <sub>REF0</sub> +0.3 <sup>Note</sup> and -0.3 to V <sub>DD</sub> +0.3 <sup>Note</sup>	V
Analog output voltage	V <sub>AO</sub>	ANO0, ANO1		-0.3 to AV <sub>REF1</sub> +0.3	V
Output current, high	I <sub>OH1</sub>	Per pin	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	-10	mA
		Total of all pins -80 mA	P00 to P04, P07, P32 to P37, P40 to P47, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	-25	mA
			P05, P06, P10 to P17, P30, P31, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117	-55	mA
	I <sub>OH2</sub>	Per pin	P20 to P27, P110, P111, P150 to P157	-0.5	mA
		Total of all pins		-2	mA
Output current, low	I <sub>OL1</sub>	Per pin	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	30	mA
		Total of all pins 200 mA	P00 to P04, P07, P32 to P37, P40 to P47, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	60	mA
			P05, P06, P10 to P17, P30, P31, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117	140	mA
	I <sub>OL2</sub>	Per pin	P20 to P27, P110, P111, P150 to P157	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T <sub>A</sub>	In normal operation mode		-40 to +85	°C
		In flash memory programming mode			
Storage temperature	T <sub>stg</sub>			-65 to +150	°C

**Note** Must be 6.5 V or lower.

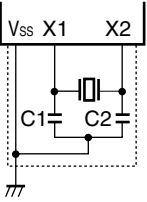
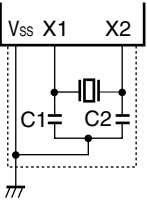
**Caution** Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.



&lt;R&gt;

**X1 Oscillator Characteristics****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )**

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency ( $f_x$ ) <sup>Note</sup>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		20.0	MHz
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	2.0		5.0	
Crystal resonator		X1 clock oscillation frequency ( $f_x$ ) <sup>Note</sup>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.0		20.0	MHz
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	2.0		5.0	

**Note** Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

**Cautions 1.** When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. Since the CPU is started by the internal high-speed oscillation clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

**Internal Oscillator Characteristics****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )**

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit	
8 MHz internal oscillator	Internal high-speed oscillation clock frequency ( $f_{IH}$ ) <sup>Note 1</sup>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	7.6	8.0	8.4	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	5.0	8.0	8.4	MHz	
240 kHz internal oscillator	Internal low-speed oscillation clock frequency ( $f_{IL}$ )	Normal current mode	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	216	240	264	kHz
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	192	240	264	kHz
		Low consumption current mode <sup>Note 2</sup>		192	240	264	kHz

**Notes 1.** This only indicates the oscillator characteristics of when HIOTRM is set to 10H. Refer to AC Characteristics for instruction execution time.

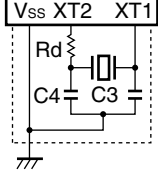
**2.** Regulator output is set to low consumption current mode in the following cases:

- When the RMC register is set to 5AH.
- During system reset.
- In STOP mode (except during OCD mode).
- When both the high-speed system clock ( $f_{MX}$ ) and the high-speed internal oscillation clock ( $f_{IH}$ ) are stopped during CPU operation with the subsystem clock ( $f_{XT}$ ).
- When both the high-speed system clock ( $f_{MX}$ ) and the high-speed internal oscillation clock ( $f_{IH}$ ) are stopped during the HALT mode when the CPU operation with the subsystem clock ( $f_{XT}$ ) has been set.

**Remark** For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 23 REGULATOR**.

&lt;R&gt;

**XT1 Oscillator Characteristics****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )**

Resonator	Recommended Circuit	Items	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency ( $f_{XT1}$ ) <sup>Note</sup>		32	32.768	35	kHz

**Note** Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

**Cautions 1.** When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.

## Recommended Oscillator Constants

(1) X1 oscillation: Ceramic resonator (AMPH = 0, RMC = 00H, T<sub>A</sub> = -40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants		Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	MIN. (V)	MAX. (V)
Murata Manufacturing Co., Ltd.	CSTCC2M00G56-R0	SMD	2.0	Internal (47)	Internal (47)	1.8	5.5
	CSTCR4M00G55-R0	SMD	4.0	Internal (39)	Internal (39)	1.8	
	CSTLS4M00G56-B0	Lead		Internal (47)	Internal (47)	1.8	
	CSTCR4M19G55-R0	SMD	4.194	Internal (39)	Internal (39)	1.8	
	CSTLS4M19G56-B0	Lead		Internal (47)	Internal (47)	1.8	
	CSTCR4M91G55-R0	SMD	4.915	Internal (39)	Internal (39)	1.8	
	CSTLS4M91G53-B0	Lead		Internal (15)	Internal (15)	1.8	
	CSTLS4M91G56-B0			Internal (47)	Internal (47)	2.1	
	CSTCR5M00G53-R0	SMD	5.0	Internal (15)	Internal (15)	1.8	
	CSTCR5M00G55-R0			Internal (39)	Internal (39)	1.8	
	CSTLS5M00G53-B0	Lead	5.0	Internal (15)	Internal (15)	1.8	
	CSTLS5M00G56-B0			Internal (47)	Internal (47)	2.1	
	CSTCR6M00G53-R0	SMD	6.0	Internal (15)	Internal (15)	1.8	
	CSTCR6M00G55-R0			Internal (39)	Internal (39)	1.9	
	CSTLS6M00G53-B0	Lead	6.0	Internal (15)	Internal (15)	1.8	
	CSTLS6M00G56-B0			Internal (47)	Internal (47)	2.2	
	CSTCE8M00G52-R0	SMD	8.0	Internal (10)	Internal (10)	1.8	
	CSTCE8M00G55-R0			Internal (33)	Internal (33)	1.9	
	CSTLS8M00G53-B0	Lead	8.0	Internal (15)	Internal (15)	1.8	
	CSTLS8M00G56-B0			Internal (47)	Internal (47)	2.4	
	CSTCE8M38G52-R0	SMD	8.388	Internal (10)	Internal (10)	1.8	
	CSTCE8M38G55-R0			Internal (33)	Internal (33)	1.9	
	CSTLS8M38G53-B0	Lead	8.388	Internal (15)	Internal (15)	1.8	
	CSTLS8M38G56-B0			Internal (47)	Internal (47)	2.4	
CSTCE10M0G52-R0	SMD	10.0	Internal (10)	Internal (10)	1.8		
CSTCE10M0G55-R0			Internal (33)	Internal (33)	2.1		
CSTLS10M0G53-B0	Lead	10.0	Internal (15)	Internal (15)	1.8		
TOKO, Inc.	DCRHTC(P)2.00LL	Lead	2.0	Internal (30)	Internal (30)	1.8	5.5
	DCRHTC(P)4.00LL		4.0	Internal (30)	Internal (30)		
	DECRHTC4.00	SMD	4.0	Internal (15)	Internal (15)		
	DCRHVC(P)8.00A	Lead	8.0	Internal (22)	Internal (22)		

**Caution** The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

<R> When doing so, check the conditions for using the AMPH bit, RMC register, and whether to enter or exit the STOP mode.

The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.

&lt;R&gt;

**(2) X1 oscillation: Crystal resonator (AMPH = 0, RMC = 00H, T<sub>A</sub> = -40 to +85°C)**

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants		Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	MIN. (V)	MAX. (V)
KYOCERA KINSEKI Co., Ltd.	HC49SFWB04194D0PPTZZ	Lead	4.194	10	10	1.8	5.5
	CX49GFWB04194D0PPTZZ						
	CX1255GB04194D0PPTZZ	SMD					
	HC49SFWB05000D0PPTZZ	Lead	5.0	10	10	1.8	
	CX49GFWB05000D0PPTZZ						
	CX1255GB05000D0PPTZZ	SMD					
	CX8045GB05000D0PPTZZ						
	HC49SFWB08380D0PPTZZ	Lead	8.38	10	10	1.8	
	CX49GFWB08380D0PPTZZ						
	CX1255GB08380D0PPTZZ	SMD					
	CX8045GB08380D0PPTZZ						
	CX5032GB08380D0PPTZZ						
	HC49SFWB10000D0PPTZZ	Lead	10.0	10	10	1.8	
	CX49GFWB10000D0PPTZZ						
	CX1255GB10000D0PPTZZ	SMD					
	CX8045GB10000D0PPTZZ						
	CX5032GB10000D0PPTZZ						
	CX5032SB10000D0PPTZZ						
CX3225GB10000D0PPTZZ							

**Caution** The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

When doing so, check the conditions for using the AMPH bit, RMC register, and whether to enter or exit the STOP mode.

The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.

**(3) X1 oscillation: Ceramic resonator (AMPH = 1, RMC = 00H, T<sub>A</sub> = -40 to +85°C)**

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants		Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	MIN. (V)	MAX. (V)
Murata Manufacturing Co., Ltd.	CSTCE12M0G55-R0	SMD	12.0	Internal (33)	Internal (33)	1.8	5.5
	CSTCE16M0V53-R0	SMD	16.0	Internal (15)	Internal (15)	1.8	
	CSTLS16M0X51-B0	Lead		Internal (5)	Internal (5)	1.8	
	CSTCE20M0V53-R0	SMD	20.0	Internal (15)	Internal (15)	1.9	
	CSTCG20M0V53-R0	Small SMD		Internal (15)	Internal (15)	2.0	
	CSTLS20M0X51-B0	Lead		Internal (5)	Internal (5)	1.9	
TOKO, Inc.	DCRHYC(P)12.00A	Lead		12.0	Internal (22)	Internal (22)	1.8
	DCRHZ(P)16.00A-15	Lead	16.0	Internal (15)	Internal (15)		
	DCRHZ(P)20.00A-15	Lead	20.0	Internal (15)	Internal (15)	2.0	
	DECRHZ20.00	SMD		Internal (10)	Internal (10)	1.8	

**Caution** The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

<R> When doing so, check the conditions for using the AMPH bit, RMC register, and whether to enter or exit the STOP mode.

The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.

&lt;R&gt;

**(4) X1 oscillation: Crystal resonator (AMPH = 1, RMC = 00H, T<sub>A</sub> = -40 to +85°C)**

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants		Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	MIN. (V)	MAX. (V)
KYOCERA KINSEKI Co., Ltd.	HC49SFWB16000D0PPTZZ	Lead	16.0	10	10	1.8	5.5
	CX49GFWB16000D0PPTZZ						
	CX1255GB16000D0PPTZZ	SMD					
	CX8045GB16000D0PPTZZ						
	CX5032GB16000D0PPTZZ						
	CX5032SB16000D0PPTZZ						
	CX3225GB16000D0PPTZZ						
	CX3225SB16000D0PPTZZ						
	CX2520SB16000D0PPTZZ						
	HC49SFWB20000D0PPTZZ						
	CX49GFWB20000D0PPTZZ						
	CX1255GB20000D0PPTZZ	SMD					
	CX8045GB20000D0PPTZZ						
	CX5032GB20000D0PPTZZ						
	CX5032SB20000D0PPTZZ						
	CX3225GB20000D0PPTZZ						
	CX3225SB20000D0PPTZZ						
	CX2520SB20000D0PPTZZ						
	CX2016SB20000D0PPTZZ						

**Caution** The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

When doing so, check the conditions for using the AMPH bit, RMC register, and whether to enter or exit the STOP mode.

The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.

**(5) XT1 oscillation: Crystal resonator ( $T_A = -40$  to  $+85^\circ\text{C}$ )**

Manufacturer	Part Number	SMD/Lead	Frequency (kHz)	Load Capacitance CL (pF)	Recommended Circuit Constants			Oscillation Voltage Range	
					C3 (pF)	C4 (pF)	Rd (k $\Omega$ )	MIN. (V)	MAX. (V)
Seiko Instruments Inc.	SP-T2A	SMD	32.768	6.0	5	5	0	1.8	5.5
				12.5	18	18	0		
	SSP-T7	Small SMD		7.0	7	7	0		
				12.5	18	18	0		
	VT-200	Lead		6.0	5	5	0		
				12.5	18	18	0		
CITIZEN FINETECH MIYOTA CO., LTD.	CM200S	SMD	32.768	9.0	12	15	0	1.8	5.5
					12	15	100		
	CM315	SMD		9.0	15	15	0		
					15	15	100		
	CM519	SMD		9.0	15	12	0		
					15	12	100		

**Caution** The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

<R> When doing so, check the conditions for using the RMC register, and whether to enter or exit the STOP mode.

The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.

**(6) XT1 oscillation: Crystal resonator ( $T_A = -20$  to  $+70^\circ\text{C}$ )**

Manufacturer	Part Number	SMD/Lead	Frequency (kHz)	Load Capacitance CL (pF)	Recommended Circuit Constants			Oscillation Voltage Range	
					C3 (pF)	C4 (pF)	Rd (k $\Omega$ )	MIN. (V)	MAX. (V)
CITIZEN FINETECH MIYOTA CO., LTD.	CFS-206	Lead	32.768	12.5	22	18	0	1.8	5.5
					22	18	100		
				9.0	12	15	0		
					12	15	100		

**Caution** The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

<R> When doing so, check the conditions for using the RMC register, and whether to enter or exit the STOP mode.

The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.



**DC Characteristics (1/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Output current, high <sup>Note 1</sup>	$I_{OH1}$	Per pin for P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-3.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-1.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-1.0	mA
		Total of P00 to P04, P07, P32 to P37, P40 to P47, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163 (When duty = 70% <sup>Note 2</sup> )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-12.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-7.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-5.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117 (When duty = 70% <sup>Note 2</sup> )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-18.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-15.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-10.0	mA
	Total of all pins (When duty = 60% <sup>Note 2</sup> )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			-23.0	mA	
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			-20.0	mA	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			-15.0	mA	
$I_{OH2}$	Per pin for P20 to P27, P150 to P157	$AV_{REF0} \leq V_{DD}$			-0.1	mA	
	Per pin for P110, P111	$AV_{REF1} \leq V_{DD}$			-0.1	mA	

**Notes** 1. Value of current at which the device operation is guaranteed even if the current flows from  $EV_{DD0}$  or  $EV_{DD1}$  pin to an output pin.

2. Specification under conditions where the duty factor is 60% or 70%.

The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins =  $(I_{OH} \times 0.7)/(n \times 0.01)$

<Example> Where  $n = 50\%$  and  $I_{OH} = -20.0\text{ mA}$

$$\text{Total output current of pins} = (-20.0 \times 0.7)/(50 \times 0.01) = -28.0\text{ mA}$$

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

**Caution** P02 to P04, P43, P45, P95 to P97, P125 to P127, and P142 to P144 do not output high level in N-ch open-drain mode.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**DC Characteristics (2/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  
 $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Output current, low <sup>Note 1</sup>	IoL1	Per pin for P00 to P02, P05 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P94, P97, P112 to P117, P120, P127, P130, P131, P140, P141, P144, P145, P160 to P163	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			8.5	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			1.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			0.5	mA
		Per pin for P03, P04, P95, P96, P125, P126, P142, P143	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			8.5	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			1.5	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			0.6	mA
		Per pin for P60 to P63	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			15.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			3.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			2.0	mA
		Total of P00 to P04, P07, P32 to P37, P40 to P47, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163 (When duty = 70% <sup>Note 2</sup> )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			20.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			15.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			9.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117 (When duty = 70% <sup>Note 2</sup> )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			45.0	mA
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			35.0	mA
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			20.0	mA
Total of all pins (When duty = 60% <sup>Note 2</sup> )	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			65.0	mA		
	$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$			50.0	mA		
	$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			29.0	mA		
IoL2	Per pin for P20 to P27, P150 to P157	$AV_{REF0} \leq V_{DD}$			0.4	mA	
	P110, P111	$AV_{REF1} \leq V_{DD}$			0.4	mA	

**Notes** 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to  $EV_{SS0}$ ,  $EV_{SS1}$ ,  $V_{SS}$ , or  $AV_{SS}$  pin.

2. Specification under conditions where the duty factor is 60% or 70%.

The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins =  $(IoL \times 0.7)/(n \times 0.01)$

<Example> Where  $n = 50\%$  and  $IoL = 20.0\text{ mA}$

$$\text{Total output current of pins} = (20.0 \times 0.7)/(50 \times 0.01) = 28.0\text{ mA}$$

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**DC Characteristics (3/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input voltage, high	$V_{IH1}$	P01, P02, P12, P13, P15, P32 to P37, P41, P45, P52 to P57, P64 to P67, P80 to P87, P90 to P94, P97, P112 to P117, P121 to P124, P127, P144	$0.7V_{DD}$		$V_{DD}$	V	
	$V_{IH2}$	P00, P03 to P06, P10, P11, P14, P16, P17, P30, P31, P40, P42 to P44, P46, P47, P50, P51, P70 to P77, P95, P96, P120, P125, P126, P131, P140 to P143, P145, P160 to P163, EXCLK, RESET	Normal input buffer	$0.8V_{DD}$		$V_{DD}$	V
	$V_{IH3}$	P03, P04, P43, P44, P95, P96, P125, P126, P142, P143	TTL input buffer $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2.2		$V_{DD}$	V
			TTL input buffer $2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	2.0		$V_{DD}$	V
			TTL input buffer $1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	1.6		$V_{DD}$	V
	$V_{IH4}$	P20 to P27, P150 to P157	$2.7\text{ V} \leq AV_{REF0} \leq V_{DD}$	$0.7AV_{REF0}$		$AV_{REF0}$	V
			$AV_{REF0} = V_{DD} < 2.7\text{ V}$				
	$V_{IH5}$	P110, P111	$2.7\text{ V} \leq AV_{REF1} \leq V_{DD}$	$0.7AV_{REF1}$		$AV_{REF1}$	V
			$AV_{REF1} = V_{DD} < 2.7\text{ V}$				
	$V_{IH6}$	P60 to P63		$0.7V_{DD}$		6.0	V
$V_{IH7}$	FLMD0		$0.9V_{DD}$ Note		$V_{DD}$	V	

**Note** Must be  $0.9V_{DD}$  or higher when used in the flash memory programming mode.

- Cautions**
- The maximum value of  $V_{IH}$  of pins P02 to P04, P43, P45, P95 to P97, P125 to P127, and P142 to P144 is  $V_{DD}$ , even in the N-ch open-drain mode.
  - For P122/EXCLK, the value of  $V_{IH}$  differ according to the input port mode or external clock mode. Make sure to satisfy the DC characteristics of EXCLK in external clock input mode.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**DC Characteristics (4/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input voltage, low	$V_{IL1}$	P01, P02, P07, P12, P13, P15, P32 to P37, P41, P45, P52 to P57, P64 to P67, P80 to P87, P90 to P94, P97, P112 to P117, P121 to P124, P127, P144	0		$0.3V_{DD}$	V	
	$V_{IL2}$	P00, P03 to P06, P10, P11, P14, P16, P17, P30, P31, P40, P42 to P44, P46, P47, P50, P51, P70 to P77, P95, P96, P120, P125, P126, P131, P140 to P143, P145, P160 to P163, EXCLK, RESET	0		$0.2V_{DD}$	V	
	$V_{IL3}$	P03, P04, P43, P44, P95, P96, P125, P126, P142, P143	TTL input buffer $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0		0.8	V
			TTL input buffer $2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$	0		0.5	V
			TTL input buffer $1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	0		0.2	V
	$V_{IL4}$	P20 to P27, P150 to P157	$2.7\text{ V} \leq AV_{REF0} \leq V_{DD}$	0		$0.3AV_{REF0}$	V
			$AV_{REF0} = V_{DD} < 2.7\text{ V}$				
	$V_{IL5}$	P110, P111	$2.7\text{ V} \leq AV_{REF1} \leq V_{DD}$	0		$0.3AV_{REF1}$	V
			$AV_{REF1} = V_{DD} < 2.7\text{ V}$				
	$V_{IL6}$	P60 to P63		0		$0.3V_{DD}$	V
$V_{IL7}$	FLMD0 <sup>Note</sup>		0		$0.1V_{DD}$	V	

**Note** When disabling writing of the flash memory, connect the FLMD0 pin processing directly to  $V_{SS}$ , and maintain a voltage less than  $0.1V_{DD}$ .

**Caution** For P122/EXCLK, the value of  $V_{IL}$  differ according to the input port mode or external clock mode. Make sure to satisfy the DC characteristics of EXCLK in external clock input mode.

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**DC Characteristics (5/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit			
Output voltage, high	V <sub>OH1</sub>	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P130, P131, P140 to P145, P160 to P163	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OH1} = -3.0\text{ mA}$	$V_{DD} - 0.7$			V		
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OH1} = -1.0\text{ mA}$	$V_{DD} - 0.5$			V		
	V <sub>OH2</sub>	P20 to P27, P150 to P157	$AV_{REF0} \leq V_{DD}$ , $I_{OH2} = -0.1\text{ mA}$	$AV_{REF0} - 0.5$			V		
		P110, P111	$AV_{REF1} \leq V_{DD}$ , $I_{OH2} = -0.1\text{ mA}$	$AV_{REF1} - 0.5$			V		
Output voltage, low	V <sub>OL1</sub>	P00 to P02, P05 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P94, P97, P112 to P117, P120, P127, P130, P131, P140, P141, P144, P145, P160 to P163	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 8.5\text{ mA}$			0.7	V		
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 1.0\text{ mA}$			0.5	V		
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 0.5\text{ mA}$			0.4	V		
			$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 8.5\text{ mA}$			0.7	V		
		P03, P04, P95, P96, P125, P126, P142, P143	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 1.5\text{ mA}$			0.5	V		
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 0.6\text{ mA}$			0.4	V		
			V <sub>OL2</sub>	P20 to P27, P150 to P157	$AV_{REF0} \leq V_{DD}$ , $I_{OL2} = 0.4\text{ mA}$			0.4	V
					P110, P111	$AV_{REF1} \leq V_{DD}$ , $I_{OL2} = 0.4\text{ mA}$			0.4
	V <sub>OL3</sub>	P60 to P63	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 15.0\text{ mA}$			2.0	V		
			$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 5.0\text{ mA}$			0.4	V		
			$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 3.0\text{ mA}$			0.4	V		
			$1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $I_{OL1} = 2.0\text{ mA}$			0.4	V		

**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**DC Characteristics (6/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Input leakage current, high	I <sub>LIH1</sub>	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P131, P140 to P145, P160 to P163, FLMD0, $\overline{\text{RESET}}$	$V_i = V_{DD}$			1	$\mu\text{A}$
	I <sub>LIH2</sub>	P20 to P27, P150 to P157	$V_i = AV_{REF0}$ , $2.7\text{ V} \leq AV_{REF0} \leq V_{DD}$			1	$\mu\text{A}$
			$V_i = AV_{REF0}$ , $AV_{REF0} = V_{DD} < 2.7\text{ V}$				
	I <sub>LIH3</sub>	P110, P111	$V_i = AV_{REF1}$ , $2.7\text{ V} \leq AV_{REF1} \leq V_{DD}$			1	$\mu\text{A}$
$V_i = AV_{REF1}$ , $AV_{REF1} = V_{DD} < 2.7\text{ V}$							
I <sub>LIH4</sub>	P121 to P124 (X1, X2, XT1, XT2)	$V_i = V_{DD}$	In Input port			1	$\mu\text{A}$
			In resonator connection			10	$\mu\text{A}$
Input leakage current, low	I <sub>LIL1</sub>	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P131, P140 to P145, P160 to P163, FLMD0, $\overline{\text{RESET}}$	$V_i = V_{SS}$			-1	$\mu\text{A}$
	I <sub>LIL2</sub>	P20 to P27, P150 to P157	$V_i = V_{SS}$ , $2.7\text{ V} \leq AV_{REF0} \leq V_{DD}$			-1	$\mu\text{A}$
			$V_i = V_{SS}$ , $AV_{REF0} = V_{DD} < 2.7\text{ V}$				
	I <sub>LIL3</sub>	P110, P111	$V_i = V_{SS}$ , $2.7\text{ V} \leq AV_{REF1} \leq V_{DD}$			-1	$\mu\text{A}$
$V_i = V_{SS}$ , $AV_{REF1} = V_{DD} < 2.7\text{ V}$							
I <sub>LIL4</sub>	P121 to P124 (X1, X2, XT1, XT2)	$V_i = V_{SS}$	In Input port			-1	$\mu\text{A}$
			In resonator connection			-10	$\mu\text{A}$

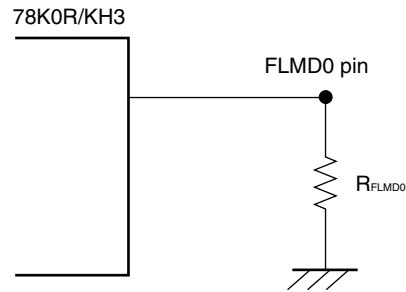
**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

**DC Characteristics (7/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Items	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
On-chip pull-up resistance	$R_U$	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P112 to P117, P120, P125 to P127, P131, P140 to P145, P160 to P163 $V_I = V_{SS}$ , In input port	10	20	100	$k\Omega$
FLMD0 pin external pull-down resistance <small>Note</small>	$R_{FLMD0}$	When enabling the self-programming mode setting with software	100			$k\Omega$

**Note** It is recommended to leave the FLMD0 pin open. If the pin is required to be pulled down externally, set  $R_{FLMD0}$  to 100  $k\Omega$  or more.



**Remark** Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

## DC Characteristics (8/12)

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit		
Supply current	$I_{DD1}$ <sup>Note 1</sup>	Operating mode	$f_{MX} = 20\text{ MHz}$ <sup>Note 2</sup> , $V_{DD} = 5.0\text{ V}$	Square wave input		7.0	12.2	mA	
				Resonator connection		7.3	12.5	mA	
			$f_{MX} = 20\text{ MHz}$ <sup>Note 2</sup> , $V_{DD} = 3.0\text{ V}$	Square wave input		7.0	12.2	mA	
				Resonator connection		7.3	12.5	mA	
			$f_{MX} = 10\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 5.0\text{ V}$	Square wave input		3.8	6.2	mA	
				Resonator connection		3.9	6.3	mA	
			$f_{MX} = 10\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 3.0\text{ V}$	Square wave input		3.8	6.2	mA	
				Resonator connection		3.9	6.3	mA	
			$f_{MX} = 5\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 3.0\text{ V}$	Normal current mode	Square wave input		2.1	3.0	mA
					Resonator connection		2.2	3.1	mA
				Low consumption current mode <sup>Note 4</sup>	Square wave input		1.5	2.1	mA
					Resonator connection		1.5	2.1	mA
			$f_{MX} = 5\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 2.0\text{ V}$	Normal current mode	Square wave input		1.4	2.1	mA
					Resonator connection		1.4	2.1	mA
Low consumption current mode <sup>Note 4</sup>	Square wave input			1.4	2.0	mA			
	Resonator connection			1.4	2.0	mA			
$f_{IH} = 8\text{ MHz}$ <sup>Note 5</sup>		$V_{DD} = 5.0\text{ V}$		3.1	5.0	mA			
		$V_{DD} = 3.0\text{ V}$		3.1	5.0	mA			

**Notes 1.** Total current flowing into  $V_{DD}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $AV_{REF0}$ , and  $AV_{REF1}$ , 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, D/A converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors.

**2.** When internal high-speed oscillator and subsystem clock are stopped.

**3.** When AMPH (bit 0 of clock operation mode control register (CMC)) = 0 and FSEL (bit 0 of operation speed mode control register (OSMC)) = 0.

**4.** When the RMC register is set to 5AH.

**5.** When high-speed system clock and subsystem clock are stopped. When FSEL (bit 0 of operation speed mode control register (OSMC)) = 0 is set.

**Remarks 1.**  $f_{MX}$ : High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

$f_{IH}$ : Internal high-speed oscillation clock frequency

**2.** For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 23 REGULATOR**.

<R> **3.** Temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$



**DC Characteristics (9/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	
Supply current	$I_{DD1}$ <sup>Note 1</sup>	Operating mode	$f_{SUB} = 32.768\text{ kHz}$ <sup>Note 2</sup> , $T_A = -40$ to $+70\text{ }^\circ\text{C}$	$V_{DD} = 5.0\text{ V}$		6.4	36.0	$\mu\text{A}$
				$V_{DD} = 3.0\text{ V}$		6.4	36.0	$\mu\text{A}$
				$V_{DD} = 2.0\text{ V}$		6.3	32.8	$\mu\text{A}$
			$f_{SUB} = 32.768\text{ kHz}$ <sup>Note 2</sup> , $T_A = -40$ to $+85\text{ }^\circ\text{C}$	$V_{DD} = 5.0\text{ V}$		6.4	51.0	$\mu\text{A}$
				$V_{DD} = 3.0\text{ V}$		6.4	51.0	$\mu\text{A}$
				$V_{DD} = 2.0\text{ V}$		6.3	47.8	$\mu\text{A}$

**Notes** 1. Total current flowing into  $V_{DD}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $AV_{REF0}$ , and  $AV_{REF1}$ , 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, D/A converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors.

2. When internal high-speed oscillator and high-speed system clock are stopped. When watchdog timer is stopped.

**Remarks** 1.  $f_{SUB}$ : Subsystem clock frequency (XT1 clock oscillation frequency)  
2. Temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$

&lt;R&gt;

**DC Characteristics (10/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit		
Supply current	$I_{DD2}$ <sup>Note 1</sup>	HALT mode	$f_{MX} = 20\text{ MHz}$ <sup>Note 2</sup> , $V_{DD} = 5.0\text{ V}$	Square wave input		1.0	2.7	mA	
				Resonator connection		1.3	3.0	mA	
			$f_{MX} = 20\text{ MHz}$ <sup>Note 2</sup> , $V_{DD} = 3.0\text{ V}$	Square wave input		1.0	2.7	mA	
				Resonator connection		1.3	3.0	mA	
			$f_{MX} = 10\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 5.0\text{ V}$	Square wave input		0.52	1.4	mA	
				Resonator connection		0.62	1.5	mA	
			$f_{MX} = 10\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 3.0\text{ V}$	Square wave input		0.52	1.4	mA	
				Resonator connection		0.62	1.5	mA	
			$f_{MX} = 5\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 3.0\text{ V}$	Normal current mode	Square wave input		0.36	0.75	mA
					Resonator connection		0.41	0.8	mA
				Low consumption current mode <sup>Note 4</sup>	Square wave input		0.22	0.5	mA
					Resonator connection		0.27	0.55	mA
			$f_{MX} = 5\text{ MHz}$ <sup>Notes 2, 3</sup> , $V_{DD} = 2.0\text{ V}$	Normal current mode	Square wave input		0.22	0.5	mA
					Resonator connection		0.27	0.55	mA
Low consumption current mode <sup>Note 4</sup>	Square wave input			0.22	0.5	mA			
	Resonator connection			0.27	0.55	mA			
$f_{IH} = 8\text{ MHz}$ <sup>Note 5</sup>		$V_{DD} = 5.0\text{ V}$		0.45	1.2	mA			
		$V_{DD} = 3.0\text{ V}$		0.45	1.2	mA			

**Notes 1.** Total current flowing into  $V_{DD}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $AV_{REF0}$ , and  $AV_{REF1}$ , including the input leakage current flowing when the level of the input pin is fixed to  $V_{DD}$  or  $V_{SS}$ . The maximum value include the peripheral operation current. However, not including the current flowing into the A/D converter, D/A converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors. During HALT instruction execution by flash memory.

2. When internal high-speed oscillator and subsystem clock are stopped.
3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0 and FSEL (bit 0 of operation speed mode control register (OSMC)) = 0.
4. When the RMC register is set to 5AH.
5. When high-speed system clock and subsystem clock are stopped. When FSEL (bit 0 of operation speed mode control register (OSMC)) = 0 is set.

**Remarks 1.**  $f_{MX}$ : High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

$f_{IH}$ : Internal high-speed oscillation clock frequency

2. For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 23 REGULATOR**.
3. Temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$

<R>

**DC Characteristics (11/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	
Supply current	$I_{DD2}$ <sup>Note 1</sup>	HALT mode	$f_{SUB} = 32.768\text{ kHz}$ <sup>Note 2</sup> , $T_A = -40$ to $+70^\circ\text{C}$	$V_{DD} = 5.0\text{ V}$		2.2	26.0	$\mu\text{A}$
				$V_{DD} = 3.0\text{ V}$		2.2	26.0	$\mu\text{A}$
				$V_{DD} = 2.0\text{ V}$		2.1	25.8	$\mu\text{A}$
			$f_{SUB} = 32.768\text{ kHz}$ <sup>Note 2</sup> , $T_A = -40$ to $+85^\circ\text{C}$	$V_{DD} = 5.0\text{ V}$		2.2	41.0	$\mu\text{A}$
				$V_{DD} = 3.0\text{ V}$		2.2	41.0	$\mu\text{A}$
				$V_{DD} = 2.0\text{ V}$		2.1	40.8	$\mu\text{A}$
	$I_{DD3}$ <sup>Note 3</sup>	STOP mode	$T_A = -40$ to $+70^\circ\text{C}$		1.1	21.0	$\mu\text{A}$	
			$T_A = -40$ to $+85^\circ\text{C}$		1.1	36.0	$\mu\text{A}$	

- Notes**
1. Total current flowing into  $V_{DD}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $AV_{REF0}$ , and  $AV_{REF1}$ , including the input leakage current flowing when the level of the input pin is fixed to  $V_{DD}$  or  $V_{SS}$ . The maximum value include the peripheral operation current. However, not including the current flowing into the A/D converter, D/A converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors. During HALT instruction execution by flash memory.
  2. When internal high-speed oscillator and high-speed system clock are stopped. When watchdog timer is stopped.
  3. Total current flowing into  $V_{DD}$ ,  $EV_{DD0}$ ,  $EV_{DD1}$ ,  $AV_{REF0}$ , and  $AV_{REF1}$ , including the input leakage current flowing when the level of the input pin is fixed to  $V_{DD}$  or  $V_{SS}$ . When subsystem clock is stopped. When watchdog timer is stopped.

- Remarks**
1.  $f_{SUB}$ : Subsystem clock frequency (XT1 clock oscillation frequency)
  2. Temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$

&lt;R&gt;

**DC Characteristics (12/12)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
RTC operating current	$I_{RTC}$ <sup>Notes 1, 2</sup>	$f_{SUB} = 32.768\text{ kHz}$	$V_{DD} = 3.0\text{ V}$		0.2	1.0	$\mu\text{A}$
			$V_{DD} = 2.0\text{ V}$		0.2	1.0	
Watchdog timer operating current	$I_{WDT}$ <sup>Notes 2, 3</sup>	$f_{IL} = 240\text{ kHz}$		5	10	$\mu\text{A}$	
A/D converter operating current	$I_{ADC}$ <sup>Note 4</sup>	During conversion at maximum speed, $2.3\text{ V} \leq AV_{REF0}$		0.86	1.9	$\text{mA}$	
D/A converter operating current	$I_{DAC}$ <sup>Note 5</sup>	Per 1 channel		1.0	2.5	$\text{mA}$	
LVI operating current	$I_{LVI}$ <sup>Note 6</sup>			9	18	$\mu\text{A}$	

- Notes**
- Current flowing only to the real-time counter (excluding the operating current of the XT1 oscillator). The current value of the 78K0R/KH3 is the TYP. value, the sum of the TYP. values of either  $I_{DD1}$  or  $I_{DD2}$ , and  $I_{RTC}$ , when the real-time counter operates in operation mode or HALT mode. The  $I_{DD1}$  and  $I_{DD2}$  MAX. values also include the real-time counter operating current.
  - When internal high-speed oscillator and high-speed system clock are stopped.
  - Current flowing only to the watchdog timer (including the operating current of the 240 kHz internal oscillator). The current value of the 78K0R/KH3 is the sum of  $I_{DD1}$ ,  $I_{DD2}$  or  $I_{DD3}$  and  $I_{WDT}$  when  $f_{CLK} = f_{SUB}/2$  or when the watchdog timer operates in STOP mode.
  - Current flowing only to the A/D converter ( $AV_{REF0}$  pin). The current value of the 78K0R/KH3 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.
  - Current flowing only to the D/A converter ( $AV_{REF1}$  pin). The current value of the 78K0R/KH3 is the sum of  $I_{DD1}$  or  $I_{DD2}$  and  $I_{DAC}$  when the D/A converter operates in an operation mode or the HALT mode.
  - Current flowing only to the LVI circuit. The current value of the 78K0R/KH3 is the sum of  $I_{DD1}$ ,  $I_{DD2}$  or  $I_{DD3}$  and  $I_{LVI}$  when the LVI circuit operates in the Operating, HALT or STOP mode.

- Remarks**
- $f_{IL}$ : Internal low-speed oscillation clock frequency  
 $f_{SUB}$ : Subsystem clock frequency (XT1 clock oscillation frequency)  
 $f_{CLK}$ : CPU/peripheral hardware clock frequency

<R> 2. Temperature condition of the TYP. value is  $T_A = 25^\circ\text{C}$

## AC Characteristics

## (1) Basic operation (1/6)

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	
Instruction cycle (minimum instruction execution time)	$t_{CY}$	Main system clock ( $f_{MAIN}$ ) operation	Normal current mode	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0.05		8	$\mu\text{s}$
				$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	0.2		8	$\mu\text{s}$
			Low consumption current mode		0.2		8	$\mu\text{s}$
		Subsystem clock ( $f_{SUB}$ ) operation			57.2	61	62.5	$\mu\text{s}$
		In the self programming mode	Normal current mode	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0.05		0.5	$\mu\text{s}$
External main system clock frequency	$f_{EX}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	Normal current mode			20.0	MHz	
			Low consumption current mode			5.0	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			2.0		5.0	MHz
External main system clock input high-level width, low-level width	$t_{EXH}, t_{EXL}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	Normal current mode			24	ns	
			Low consumption current mode			96	ns	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			96		ns	
TI00 to TI07 input high-level width, low-level width	$t_{TIH}, t_{TIL}$			$1/f_{MCK} + 10$			ns	
TO00 to TO07 output frequency	$f_{TO}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$				10	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$				5	MHz	
PCLBUZ0, PCLBUZ1 output frequency	$f_{PCL}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$				10	MHz	
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$				5	MHz	
Interrupt input high-level width, low-level width	$t_{INTH}, t_{INTL}$			1			$\mu\text{s}$	
Key interrupt input low-level width	$t_{KR}$			250			ns	
$\overline{\text{RESET}}$ low-level width	$t_{RSL}$			10			$\mu\text{s}$	

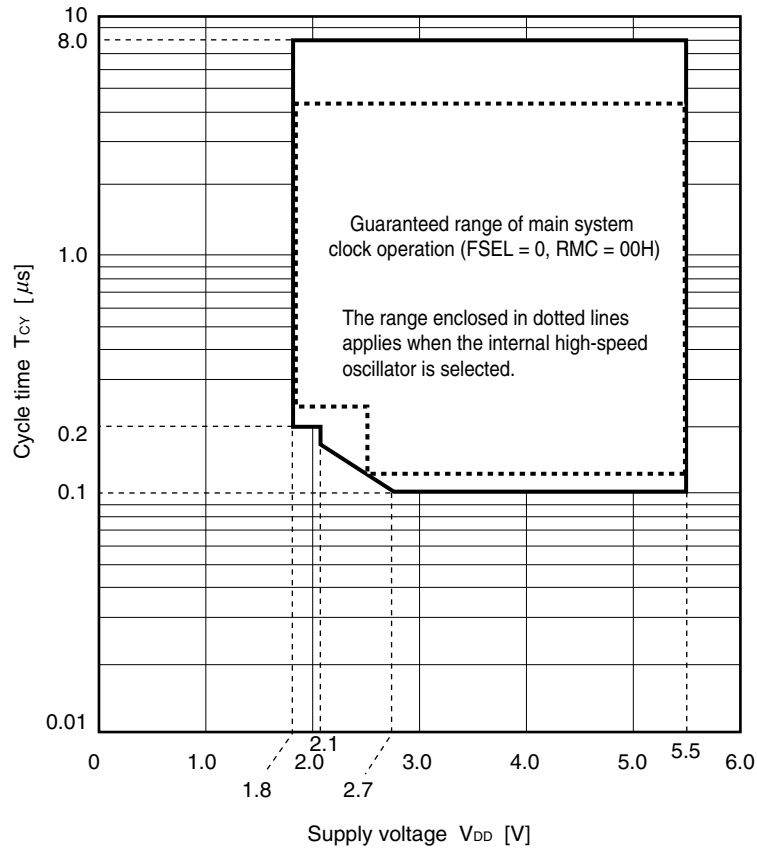
**Remarks 1.**  $f_{MCK}$ : Timer array unit operation clock frequency

(Operation clock to be set by the CKS0n bit of the TMR0n register. n: Channel number (n = 0 to 7))

- 2.** For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 23 REGULATOR**.

## (1) Basic operation (2/6)

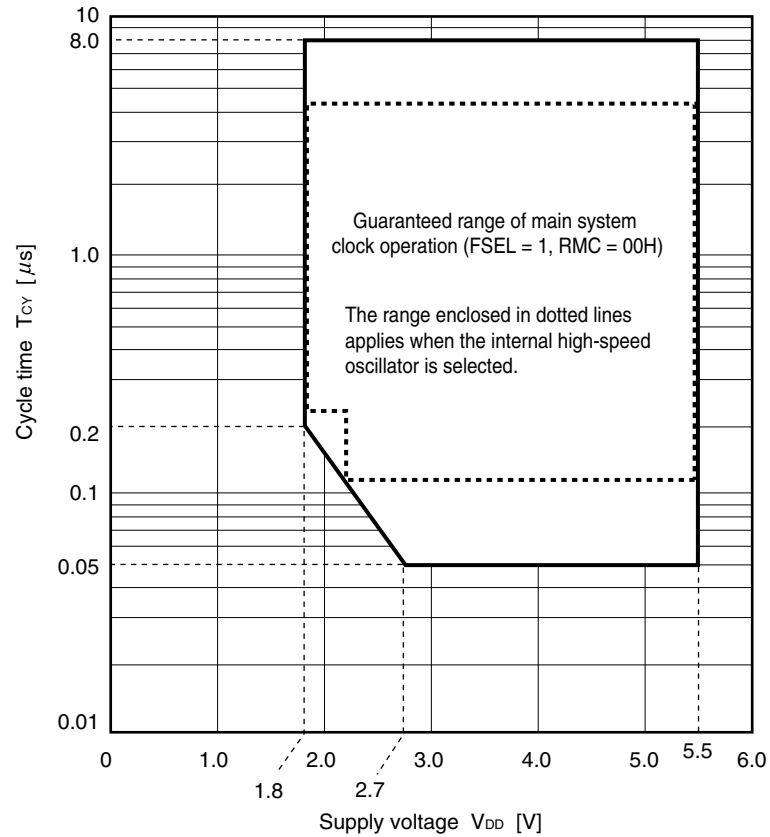
Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 00H)



**Remark** FSEL: Bit 0 of the operation speed mode control register (OSMC)  
 RMC: Regulator mode control register

## (1) Basic operation (3/6)

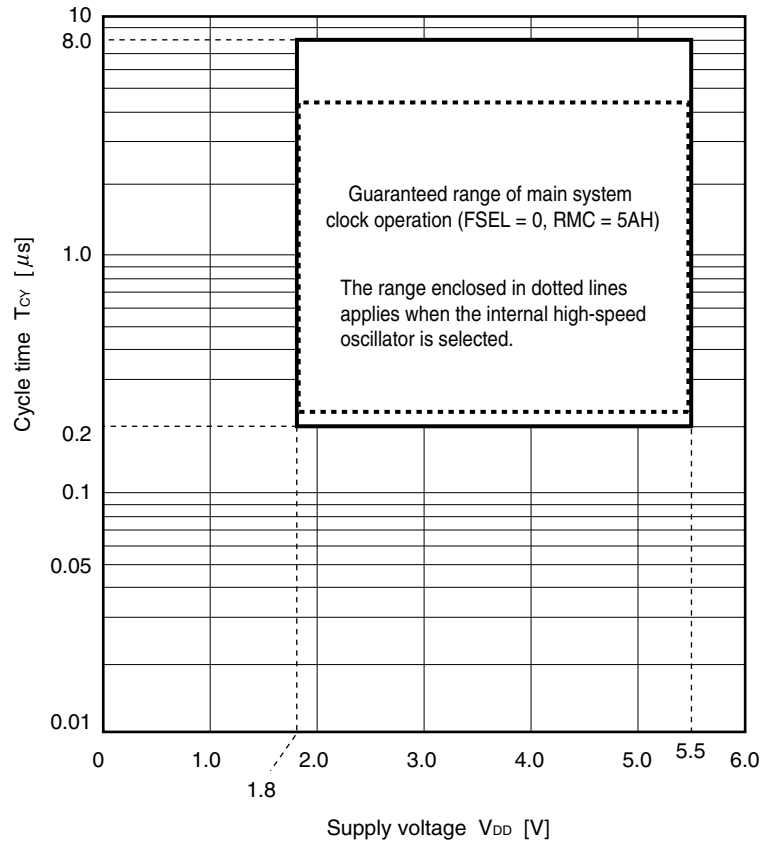
## Minimum instruction execution time during main system clock operation (FSEL = 1, RMC = 00H)



**Remark** FSEL: Bit 0 of the operation speed mode control register (OSMC)  
 RMC: Regulator mode control register

## (1) Basic operation (4/6)

## Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 5AH)

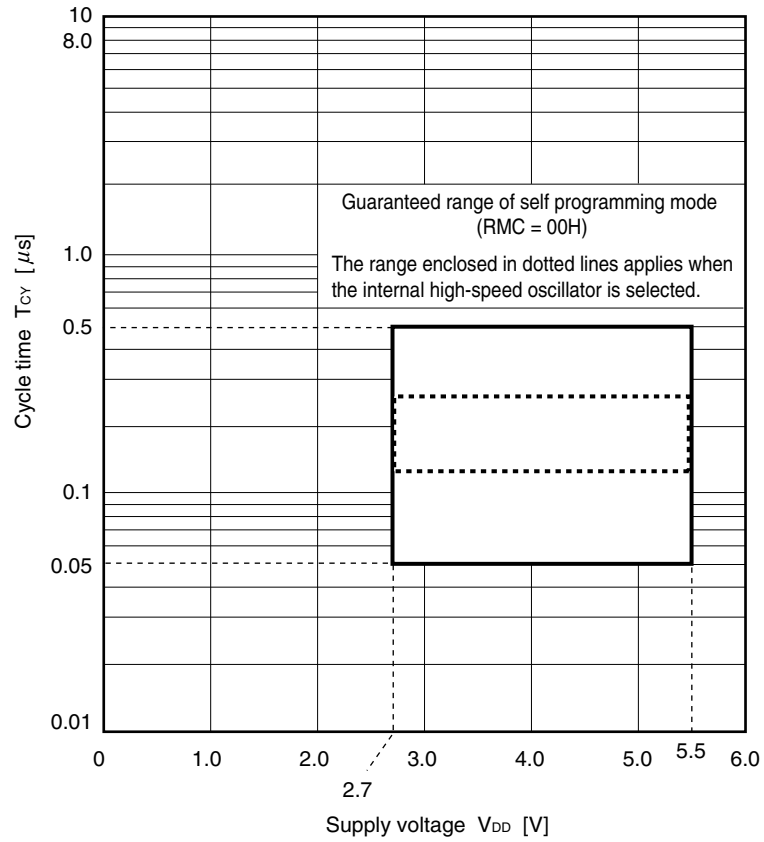


- Remarks**
1. FSEL: Bit 0 of the operation speed mode control register (OSMC)  
RMC: Regulator mode control register
  2. The entire voltage range is 5 MHz (MAX.) when RMC is set to 5AH.



## (1) Basic operation (5/6)

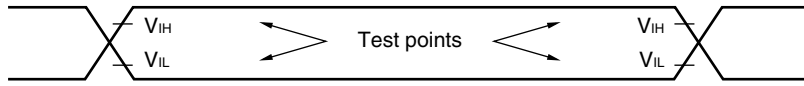
## Minimum instruction execution time during self programming mode (RMC = 00H)



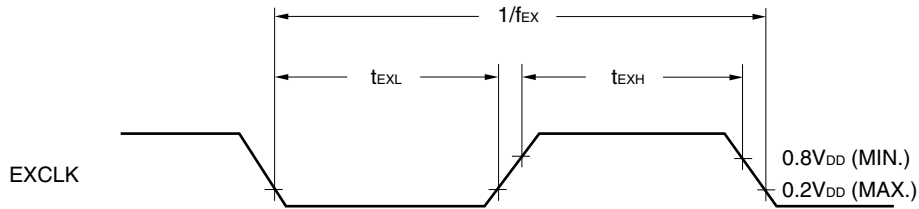
- Remarks**
1. RMC: Regulator mode control register
  2. The self programming function cannot be used when RMC is set to 5AH or the CPU operates with the subsystem clock.

(1) Basic operation (6/6)

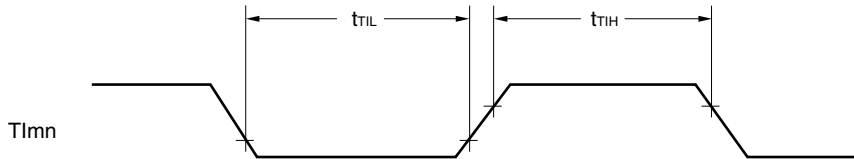
AC Timing Test Points (Excluding external bus interface)



External Main System Clock Timing

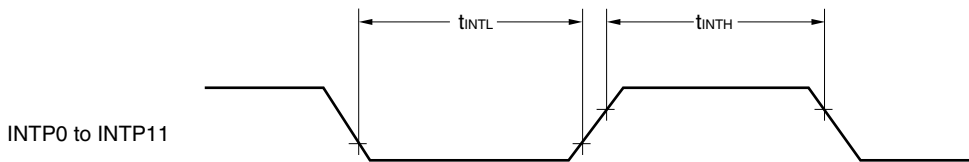


TI Timing

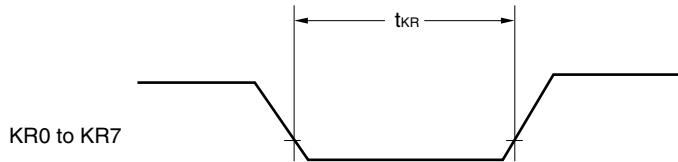


**Remark** n: Channel number, m: Unit number, mn = 00 to 07, 10 to 13

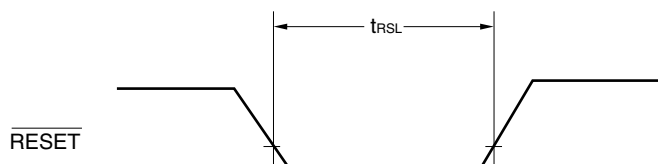
Interrupt Request Input Timing



Key Interrupt Input Timing



RESET Input Timing



## (2) External bus interface (1/4)

 $(T_A = -40 \text{ to } +85^\circ\text{C}, 1.8 \text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5 \text{ V}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 \text{ V})$ 

## (a) Read/write cycle (CLKOUT synchronous)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CLKOUT cycle	$t_{CYK}$	<1>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	100		ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	200		ns
CLKOUT high-level width	$t_{WKH}$	<2>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$0.4t_{CYK} - 30$	$0.6t_{CYK}$	ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	$0.4t_{CYK} - 50$	$0.6t_{CYK}$	ns
CLKOUT low-level width	$t_{WKL}$	<3>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$0.4t_{CYK} - 30$	$0.6t_{CYK}$	ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	$0.4t_{CYK} - 50$	$0.6t_{CYK}$	ns
ASTB high-level width	$t_{WASH1}$	<4>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$0.8t_{CYK} - 40$	$1.2t_{CYK}$	ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	$0.8t_{CYK} - 60$	$1.2t_{CYK}$	ns
$\overline{RD}$ low-level width	$t_{WRDL1}$	<5>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$(0.8 + m + w) t_{CYK} - 40$	$(1.2 + m + w) t_{CYK}$	ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	$(0.8 + m + w) t_{CYK} - 60$	$(1.2 + m + w) t_{CYK}$	ns
$\overline{WR0}, \overline{WR1}$ low-level width	$t_{WWRL1}$	<6>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$(0.8 + w) t_{CYK} - 40$	$(1.2 + w) t_{CYK}$	ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	$(0.8 + w) t_{CYK} - 60$	$(1.2 + w) t_{CYK}$	ns
Delay time from CLKOUT $\uparrow$ to ASTB	$t_{DKAS1}$	<7>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	2	50	ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	2	80	ns
Delay time from CLKOUT $\uparrow$ to $\overline{RD}$	$t_{DKRD1}$	<8>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	2	50	ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	2	80	ns
Delay time from CLKOUT $\uparrow$ to $\overline{WR0}, \overline{WR1}$	$t_{DKWR1}$	<9>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	2	50	ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	2	80	ns
Delay time from CLKOUT $\uparrow$ to address	$t_{DKA1}$	<10>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	2	55	ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	2	90	ns
Data output delay time from CLKOUT $\uparrow$	$t_{DKOD1}$	<11>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	2	55	ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	2	90	ns
Data output hold time from CLKOUT $\uparrow$	$t_{HKOD1}$	<12>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	2		ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	2		ns
Data input setup time to CLKOUT $\uparrow$	$t_{SKD11}$	<13>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	40		ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	85		ns
Data input hold time from CLKOUT $\uparrow$	$t_{HKD11}$	<14>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0		ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	0		ns
$\overline{WAIT}$ setup time to CLKOUT $\uparrow$	$t_{SKWT1}$	<15>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	45		ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	85		ns
$\overline{WAIT}$ hold time from CLKOUT $\uparrow$	$t_{HKWT1}$	<16>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0		ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	0		ns
Delay time from address output stop to $\overline{RD}\downarrow$	$t_{DAR1}$	<17>	$2.7 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	0		ns
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	0		ns

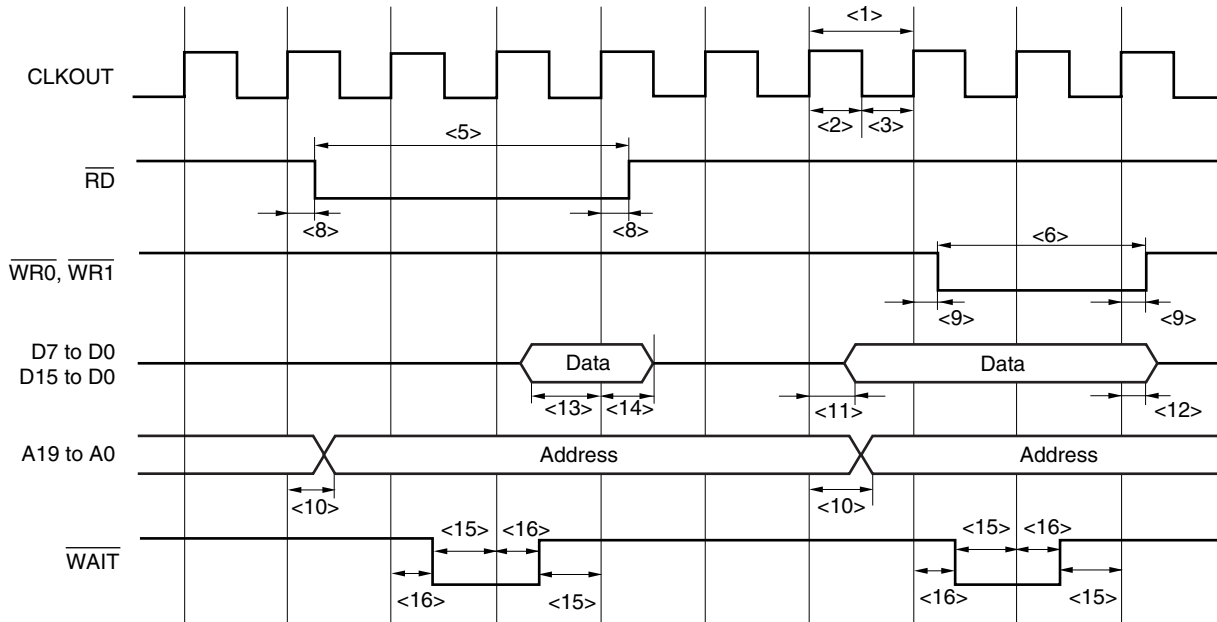
**Remarks 1.** CL: The pin load capacitance is 15 pF.**2.** Test points:  $V_{OH} = 0.8V_{DD}$ ,  $V_{OL} = 0.2V_{DD}$ **3.** m = 0: Multiplexed bus mode

m = 1: Separate bus mode

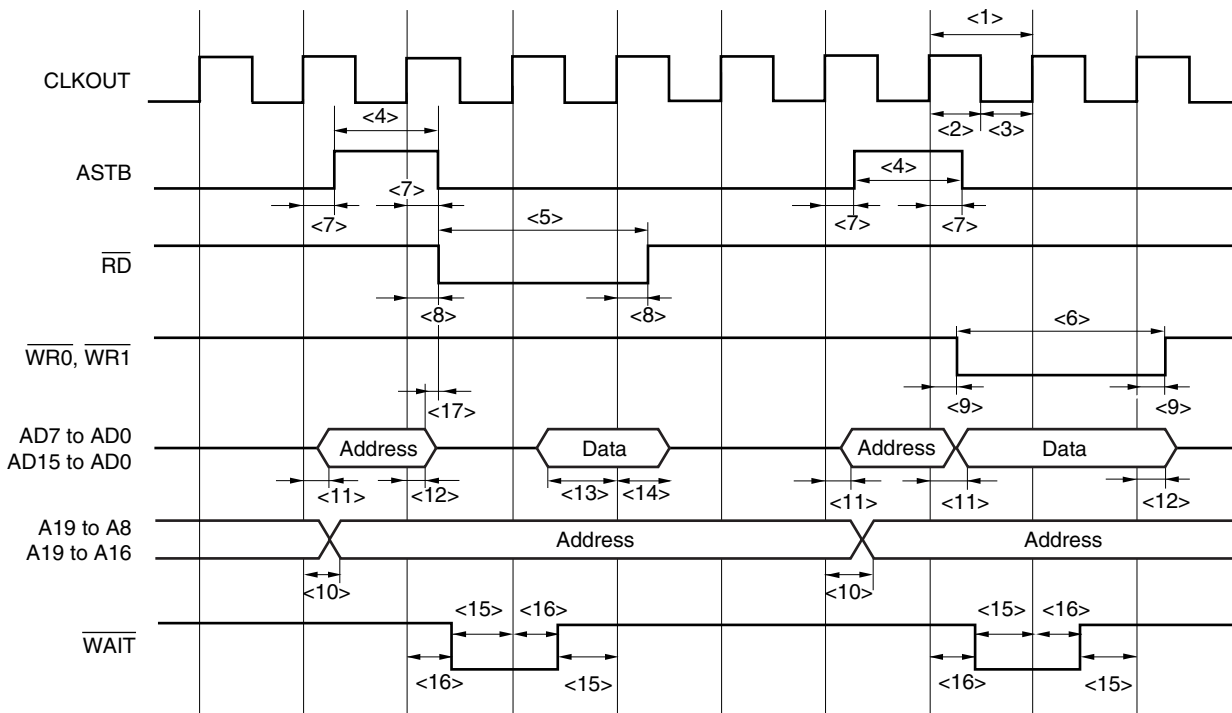
w: Number of waits with  $\overline{WAIT}$

(2) External bus interface (2/4)

Read/write cycle (CLKOUT synchronous): In separate bus mode



Read/write cycle (CLKOUT synchronous): In multiplexed bus mode



**(2) External bus interface (3/4)****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )****(b) Read/write cycle (CLKOUT asynchronous)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CLKOUT cycle	$t_{CYK2}$	<18>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	100		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	200		ns
$\overline{RD}$ low-level width	$t_{WRDL2}$	<19>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$1.8t_{CYK2} - 40$	$2.2t_{CYK2}$	ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$1.8t_{CYK2} - 60$	$2.2t_{CYK2}$	ns
$\overline{WR0}$ , $\overline{WR1}$ low-level width	$t_{WRL2}$	<20>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$0.8t_{CYK2} - 40$	$1.2t_{CYK2}$	ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$0.8t_{CYK2} - 60$	$1.2t_{CYK2}$	ns
Data input setup time to $\overline{RD}\uparrow$	$t_{SRDDI2}$	<21>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	90		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	170		ns
Data input hold time from $\overline{RD}\uparrow$	$t_{HRDDI2}$	<22>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	0		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	0		ns
Data output setup time to $\overline{WR0}$ , $\overline{WR1}\downarrow$	$t_{SWROD2}$	<23>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$t_{CYK2} - 5$		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$t_{CYK2} - 15$		ns
Data output hold time from $\overline{WR0}$ , $\overline{WR1}\uparrow$	$t_{HKOD2}$	<24>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	2		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	2		ns
Delay time from $\overline{RD}\downarrow$ to address	$t_{DRDA2}$	<25>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$		5	ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$		15	ns
Address setup time to $\overline{WR0}$ , $\overline{WR1}\downarrow$	$t_{SWRA2}$	<26>	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	$t_{CYK2} - 5$		ns
			$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$	$t_{CYK2} - 15$		ns

**Cautions** 1. CLKOUT output is not used during CLKOUT asynchronous operation, but a CPU wait occurs according to the setting of bits 4 and 5 (EW0, EW1) of the memory expansion mode control register (MEM). When  $f_{CLK}$  is sufficiently high, insert a wait by setting the EW0 and EW1 bits.

2. Do not use the WAIT pin during CLKOUT asynchronous operation.

Use the separate bus mode during CLKOUT asynchronous operation.

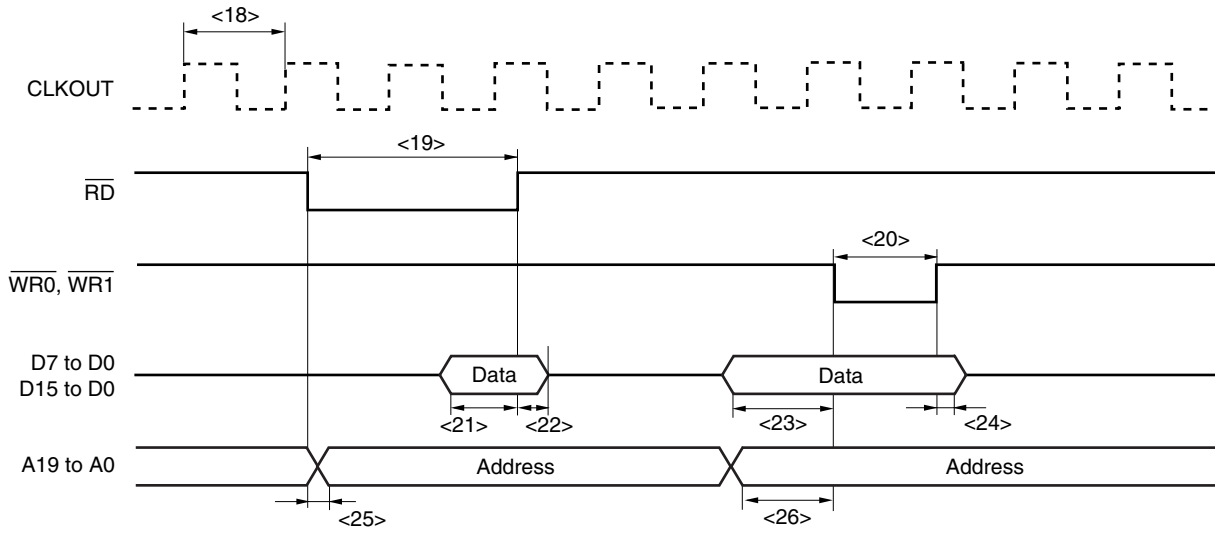
**Remarks** 1.  $f_{CLK}$ : CPU/peripheral hardware clock frequency

2. CL: The pin load capacitance is 15 pF.

3. Test points:  $V_{OH} = 0.8V_{DD}$ ,  $V_{OL} = 0.2V_{DD}$

(2) External bus interface (4/4)

Read/write cycle (CLKOUT asynchronous): In separate bus mode



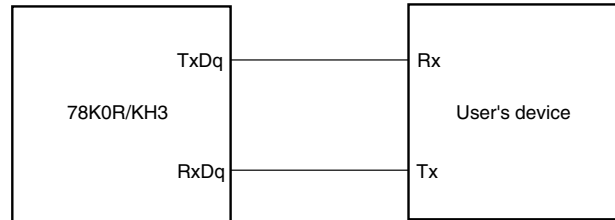
(3) Serial interface: Serial array unit (1/18)

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

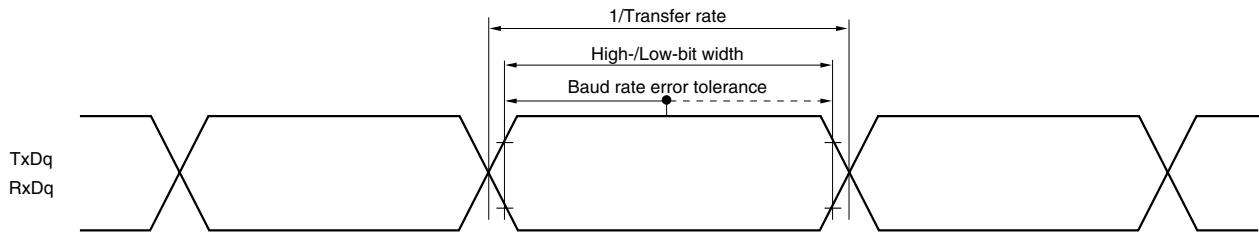
(a) During communication at same potential (UART mode) (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					$f_{MCK}/6$	bps
		$f_{CLK} = 20\text{ MHz}$ , $f_{MCK} = f_{CLK}$			3.3	Mbps

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



**Caution** Select the normal input buffer for RxDi and the normal output mode for TxDi by using the PIMg and POMg registers.

- Remarks**
- q: UART number (q = 0 to 3), g: PIM and POM number (g = 0, 14),  
i: UART number for which communication at different potential can be selected (i = 1, 2)
  - $f_{MCK}$ : Serial array unit operation clock frequency  
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1),  
n: Channel number (n = 0 to 3))

**(3) Serial interface: Serial array unit (2/18)** $(T_A = -40 \text{ to } +85^\circ\text{C}, 1.8 \text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5 \text{ V}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 \text{ V})$ <R> **(b) During communication at same potential (CSI mode) (master mode,  $\overline{\text{SCKp}}$ ... internal clock output)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	$t_{\text{KCY1}}$	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	200 <sup>Note 1</sup>			ns
		$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	300 <sup>Note 1</sup>			ns
		$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	600 <sup>Note 1</sup>			ns
$\overline{\text{SCKp}}$ high-/low-level width	$t_{\text{KH1}},$ $t_{\text{KL1}}$	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$t_{\text{KCY1}}/2 - 20$			ns
		$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	$t_{\text{KCY1}}/2 - 35$			ns
		$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	$t_{\text{KCY1}}/2 - 80$			ns
Slp setup time (to $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 2</sup>	$t_{\text{SIK1}}$	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	70			ns
		$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	100			ns
		$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	190			ns
Slp hold time (from $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 3</sup>	$t_{\text{KSH1}}$		30			ns
Delay time from $\overline{\text{SCKp}}\downarrow$ to SOp output <sup>Note 4</sup>	$t_{\text{KSO1}}$	$C = 30 \text{ pF}$ <sup>Note 5</sup>			40	ns

**Notes** 1. The value must also be  $4/f_{\text{CLK}}$  or more.2. When  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 0$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 1$ . The Slp setup time becomes “to  $\overline{\text{SCKp}}\downarrow$ ” when  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 1$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 0$ .3. When  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 0$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 1$ . The Slp hold time becomes “from  $\overline{\text{SCKp}}\downarrow$ ” when  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 1$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 0$ .4. When  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 0$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 1$ . The delay time to SOp output becomes “from  $\overline{\text{SCKp}}\uparrow$ ” when  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 1$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 0$ .5. C is the load capacitance of the  $\overline{\text{SCKp}}$  and SOp output lines.**Caution** Select the normal input buffer for Slj and the normal output mode for SOj and  $\overline{\text{SCKj}}$  by using the PIMg and POMg registers.**Remarks** 1. p: CSI number (p = 00, 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14),

j: CSI number for which communication at different potential can be selected (j = 01, 10, 11, 20, 21)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)



**(3) Serial interface: Serial array unit (3/18)** $(T_A = -40 \text{ to } +85^\circ\text{C}, 1.8 \text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5 \text{ V}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 \text{ V})$ 

&lt;R&gt;

**(c) During communication at same potential (CSI mode) (slave mode,  $\overline{\text{SCKp}}$ ... external clock input)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
$\overline{\text{SCKp}}$ cycle time	$t_{KCY2}$	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$	$6/f_{MCK}$			ns	
		$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}$	$16 \text{ MHz} < f_{MCK}$	$8/f_{MCK}$			ns
			$f_{MCK} \leq 16 \text{ MHz}$	$6/f_{MCK}$			ns
		$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$	$16 \text{ MHz} < f_{MCK}$	$8/f_{MCK}$			ns
			$f_{MCK} \leq 16 \text{ MHz}$	$6/f_{MCK}$			ns
$\overline{\text{SCKp}}$ high-/low-level width	$t_{KH2}$ , $t_{KL2}$		$f_{KCY2}/2$			ns	
Slp setup time (to $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 1</sup>	$t_{SIK2}$		80			ns	
Slp hold time (from $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 2</sup>	$t_{KSI2}$		$1/f_{MCK} + 50$			ns	
Delay time from $\overline{\text{SCKp}}\downarrow$ to SO <sub>p</sub> output <sup>Note 3</sup>	$t_{KSO2}$	$C = 30 \text{ pF}$ <sup>Note 4</sup>	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}$		$2/f_{MCK} + 45$	ns	
			$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}$		$2/f_{MCK} + 57$	ns	
			$1.8 \text{ V} \leq V_{DD} < 2.7 \text{ V}$		$2/f_{MCK} + 125$	ns	

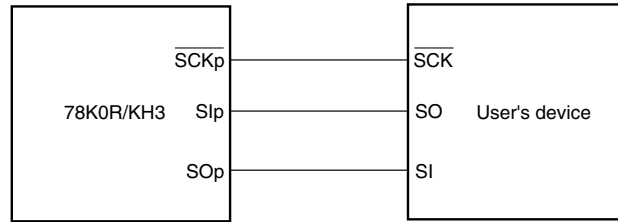
- Notes**
- When  $DAP_{mn} = 0$  and  $CKP_{mn} = 0$ , or  $DAP_{mn} = 1$  and  $CKP_{mn} = 1$ . The Slp setup time becomes “to  $\overline{\text{SCKp}}\downarrow$ ” when  $DAP_{mn} = 0$  and  $CKP_{mn} = 1$ , or  $DAP_{mn} = 1$  and  $CKP_{mn} = 0$ .
  - When  $DAP_{mn} = 0$  and  $CKP_{mn} = 0$ , or  $DAP_{mn} = 1$  and  $CKP_{mn} = 1$ . The Slp hold time becomes “from  $\overline{\text{SCKp}}\downarrow$ ” when  $DAP_{mn} = 0$  and  $CKP_{mn} = 1$ , or  $DAP_{mn} = 1$  and  $CKP_{mn} = 0$ .
  - When  $DAP_{mn} = 0$  and  $CKP_{mn} = 0$ , or  $DAP_{mn} = 1$  and  $CKP_{mn} = 1$ . The delay time to SO<sub>p</sub> output becomes “from  $\overline{\text{SCKp}}\uparrow$ ” when  $DAP_{mn} = 0$  and  $CKP_{mn} = 1$ , or  $DAP_{mn} = 1$  and  $CKP_{mn} = 0$ .
  - C is the load capacitance of the and SO<sub>p</sub> output lines.

**Caution** Select the normal input buffer for SI<sub>j</sub> and  $\overline{\text{SCKj}}$  and the normal output mode for SO<sub>j</sub> by using the PIM<sub>g</sub> and POM<sub>g</sub> registers.

- Remarks**
- p: CSI number (p = 00, 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14),  
j: CSI number for which communication at different potential can be selected (j = 01, 10, 11, 20, 21)
  - $f_{MCK}$ : Serial array unit operation clock frequency  
(Operation clock to be set by the CKS<sub>mn</sub> bit of the SMR<sub>mn</sub> register. m: Unit number (m = 0, 1),  
n: Channel number (n = 0 to 3))

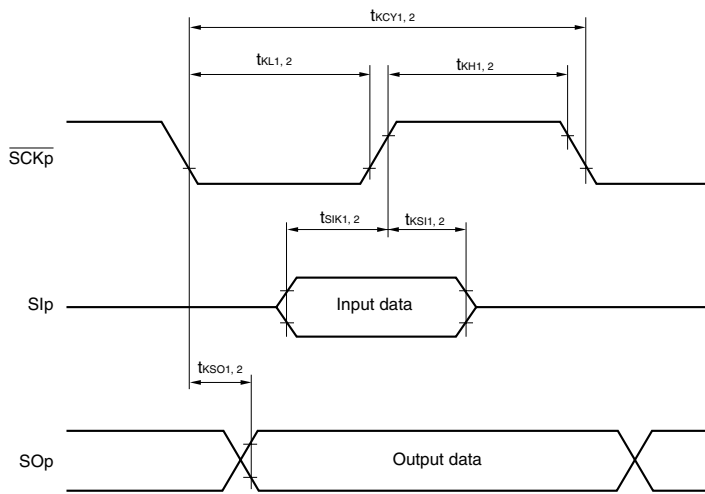
(3) Serial interface: Serial array unit (4/18)

CSI mode connection diagram (during communication at same potential)



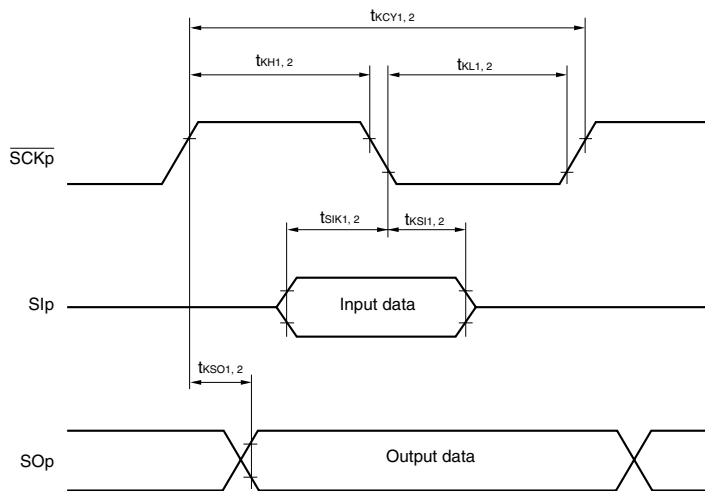
CSI mode serial transfer timing (during communication at same potential)

(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



CSI mode serial transfer timing (during communication at same potential)

(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



- Remarks**
1. p: CSI number (p = 00, 01, 10, 11, 20, 21)
  2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

## (3) Serial interface: Serial array unit (5/18)

(T<sub>A</sub> = -40 to +85°C, 1.8 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)(d) During communication at same potential (simplified I<sup>2</sup>C mode)

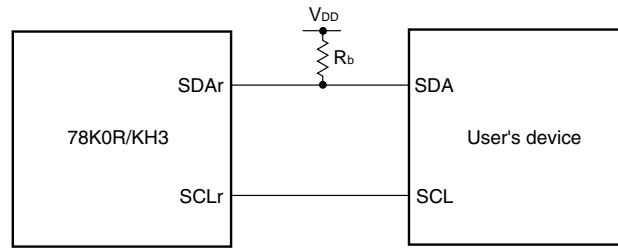
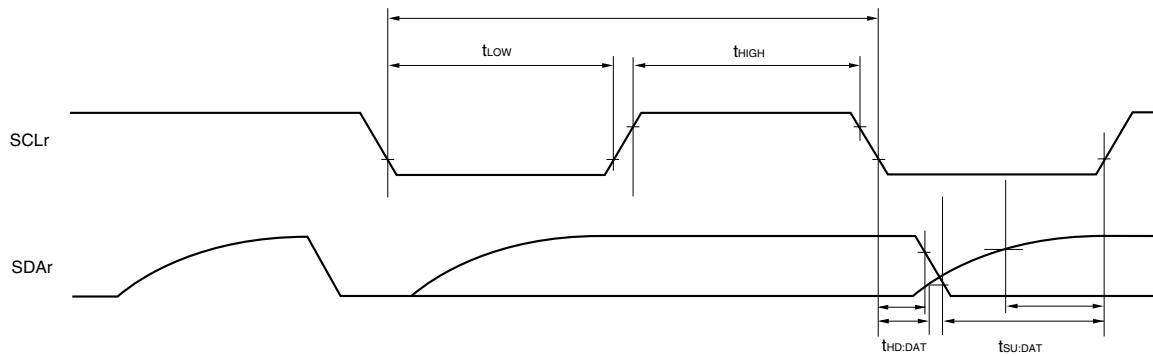
&lt;R&gt;

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f <sub>SCL</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ		400 <sup>Note</sup>	kHz
		1.8 V ≤ V <sub>DD</sub> < 2.7 V C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ		300 <sup>Note</sup>	kHz
Hold time when SCLr = "L"	t <sub>LOW</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	995		ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	1500		ns
Hold time when SCLr = "H"	t <sub>HIGH</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	995		ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	1500		ns
Data setup time (reception)	t <sub>SU:DAT</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	1/f <sub>MCK</sub> + 120		ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	1/f <sub>MCK</sub> + 230		ns
Data hold time (transmission)	t <sub>HD:DAT</sub>	2.7 V ≤ V <sub>DD</sub> ≤ 5.5 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 3 kΩ	0	160	ns
		1.8 V ≤ V <sub>DD</sub> < 2.7 V C <sub>b</sub> = 100 pF, R <sub>b</sub> = 5 kΩ	0	210	ns

&lt;R&gt;

**Note** The value must also be f<sub>MCK</sub>/4 or less.

## (3) Serial interface: Serial array unit (6/18)

Simplified I<sup>2</sup>C mode connection diagram (during communication at same potential)Simplified I<sup>2</sup>C mode serial transfer timing (during communication at same potential)

**Caution** Select the normal input buffer and the N-ch open-drain output (V<sub>DD</sub> tolerance) mode for SDAr and the normal output mode for SCLr by using the PIMg and POMg registers.

- Remarks**
1. R<sub>b</sub>[Ω]: Communication line (SDAr) pull-up resistance,  
C<sub>b</sub>[F]: Communication line (SCLr, SDAr) load capacitance
  2. r: IIC number (r = 10, 11, 20, 21), g: PIM and POM number (g = 0, 9, 12, 14)
  3. f<sub>MCK</sub>: Serial array unit operation clock frequency  
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1),  
n: Channel number (n = 0, 3), mn = 02, 03, 10, 11)

**(3) Serial interface: Serial array unit (7/18)****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )****(e) During communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output) (1/2)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Transfer rate		Reception	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,			$f_{MCK}/6$	bps
			$2.7\text{ V} \leq V_b \leq 4.0\text{ V}$	$f_{CLK} = 20\text{ MHz}$ , $f_{MCK} = f_{CLK}$			3.3
			$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$ ,			$f_{MCK}/6$	bps
			$2.3\text{ V} \leq V_b \leq 2.7\text{ V}$	$f_{CLK} = 20\text{ MHz}$ , $f_{MCK} = f_{CLK}$			3.3

**Caution** Select the TTL input buffer for RxDq and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for TxDq by using the PIMg and POMg registers.

**Remarks 1.** q: UART number (q = 1, 2), g: PIM and POM number (g = 0, 14)

**2.**  $f_{MCK}$ : Serial array unit operation clock frequency

(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1),

n: Channel number (n = 0 to 3))

**3.**  $V_{IH}$  and  $V_{IL}$  below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.

$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$ :  $V_{IH} = 2.2\text{ V}$ ,  $V_{IL} = 0.8\text{ V}$

$2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}$ ,  $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$ :  $V_{IH} = 2.0\text{ V}$ ,  $V_{IL} = 0.5\text{ V}$

**4.** UART0 and UART3 cannot communicate at different potential. Use UART1 and UART2 for communication at different potential.

**(3) Serial interface: Serial array unit (8/18)** $(T_A = -40 \text{ to } +85^\circ\text{C}, 2.7 \text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5 \text{ V}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 \text{ V})$ **(e) During communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output) (2/2)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Transfer rate		Transmission	4.0 V $\leq$ V <sub>DD</sub> $\leq$ 5.5 V, 2.7 V $\leq$ V <sub>b</sub> $\leq$ 4.0 V			<b>Note 1</b>	
			$f_{CLK} = 16.8 \text{ MHz}, f_{MCK} = f_{CLK},$ $C_b = 50 \text{ pF}, R_b = 1.4 \text{ k}\Omega, V_b = 2.7 \text{ V}$			2.8 <sup>Note 2</sup>	Mbps
			2.7 V $\leq$ V <sub>DD</sub> < 4.0 V, 2.3 V $\leq$ V <sub>b</sub> $\leq$ 2.7 V			<b>Note 3</b>	
			$f_{CLK} = 19.2 \text{ MHz}, f_{MCK} = f_{CLK},$ $C_b = 50 \text{ pF}, R_b = 2.7 \text{ k}\Omega, V_b = 2.3 \text{ V}$			1.2 <sup>Note 4</sup>	Mbps

**Notes** 1. The smaller maximum transfer rate derived by using  $f_{MCK}/6$  or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V  $\leq$  V<sub>DD</sub> = EV<sub>DD</sub>  $\leq$  5.5 V and 2.7 V  $\leq$  V<sub>b</sub>  $\leq$  4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

\* This value is the theoretical value of the relative difference between the transmission and reception sides.

- This value as an example is calculated when the conditions described in the "Conditions" column are met. Refer to Note 1 above to calculate the maximum transfer rate under conditions of the customer.
- The smaller maximum transfer rate derived by using  $f_{MCK}/6$  or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V  $\leq$  V<sub>DD</sub> = EV<sub>DD</sub> < 4.0 V and 2.3 V  $\leq$  V<sub>b</sub>  $\leq$  2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [%]}$$

\* This value is the theoretical value of the relative difference between the transmission and reception sides.

- This value as an example is calculated when the conditions described in the "Conditions" column are met. Refer to Note 3 above to calculate the maximum transfer rate under conditions of the customer.

**Caution** Select the TTL input buffer for RxDq and the N-ch open-drain output (V<sub>DD</sub> tolerance) mode for TxDq by using the PIMg and POMg registers.

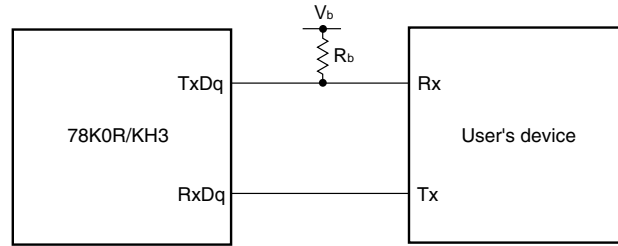
(Remarks are given on the next page.)

**(3) Serial interface: Serial array unit (9/18)**

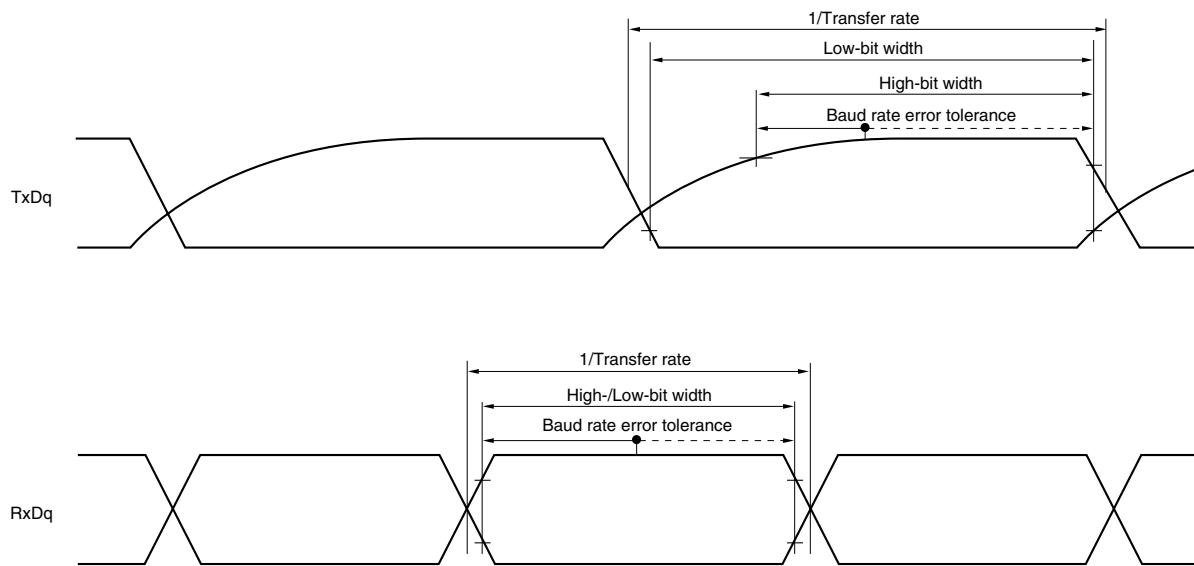
- Remarks**
1.  $R_b[\Omega]$ : Communication line (TxDq) pull-up resistance,  
 $C_b[F]$ : Communication line (TxDq) load capacitance,  $V_b[V]$ : Communication line voltage
  2. q: UART number (q = 1, 2), g: PIM and POM number (g = 0, 14)
  3.  $f_{MCK}$ : Serial array unit operation clock frequency  
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1),  
n: Channel number (n = 0 to 3))
  4.  $V_{IH}$  and  $V_{IL}$  below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.  
 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$ :  $V_{IH} = 2.2\text{ V}$ ,  $V_{IL} = 0.8\text{ V}$   
 $2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}$ ,  $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$ :  $V_{IH} = 2.0\text{ V}$ ,  $V_{IL} = 0.5\text{ V}$
  5. UART0 and UART3 cannot communicate at different potential. Use UART1 and UART2 for communication at different potential.

## (3) Serial interface: Serial array unit (10/18)

## UART mode connection diagram (during communication at different potential)



## UART mode bit width (during communication at different potential) (reference)



**Caution** Select the TTL input buffer for RxDq and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for TxDq by using the PIMg and POMg registers.

- Remarks**
1.  $R_b[\Omega]$ : Communication line (TxDq) pull-up resistance,  $V_b[V]$ : Communication line voltage
  2. q: UART number (q = 1, 2), g: PIM and POM number (g = 0, 14)
  3. UART0 and UART3 cannot communicate at different potential. Use UART1 and UART2 for communication at different potential.



## (3) Serial interface: Serial array unit (11/18)

(T<sub>A</sub> = -40 to +85°C, 2.7 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)<R> (f) During communication at different potential (2.5 V, 3 V) (CSI mode) (master mode,  $\overline{\text{SCKp}}$ ... internal clock output) (1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	t <sub>KCY1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	400 <sup>Note 1</sup>			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	800 <sup>Note 1</sup>			ns
$\overline{\text{SCKp}}$ high-level width	t <sub>KH1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	t <sub>KCY1</sub> /2 - 75			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	t <sub>KCY1</sub> /2 - 170			ns
$\overline{\text{SCKp}}$ low-level width	t <sub>KL1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	t <sub>KCY1</sub> /2 - 20			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	t <sub>KCY1</sub> /2 - 35			ns
Slp setup time (to $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 2</sup>	t <sub>SIK1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	150			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	275			ns
Slp hold time (from $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 2</sup>	t <sub>KSH1</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ	30			ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ	30			ns
Delay time from $\overline{\text{SCKp}}\downarrow$ to SOp output <sup>Note 2</sup>	t <sub>KSO1</sub>	4.0 V ≤ V <sub>DD</sub> < 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 1.4 kΩ			120	ns
		2.7 V ≤ V <sub>DD</sub> < 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 30 pF, R <sub>b</sub> = 2.7 kΩ			215	ns

**Notes 1.** The value must also be 4/f<sub>CLK</sub> or more.2. When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 0, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 1.**Caution** Select the TTL input buffer for Slp and the N-ch open-drain output (V<sub>DD</sub> tolerance) mode for SOp and  $\overline{\text{SCKp}}$  by using the PIMg and POMg registers.**Remarks 1.** p: CSI number (p = 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14)

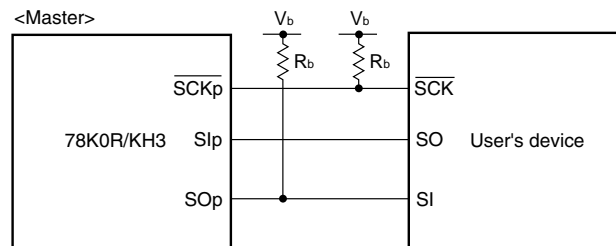
2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

3. R<sub>b</sub>[Ω]: Communication line ( $\overline{\text{SCKp}}$ , SOp) pull-up resistance,C<sub>b</sub>[F]: Communication line ( $\overline{\text{SCKp}}$ , SOp) load capacitance, V<sub>b</sub>[V]: Communication line voltage4. V<sub>IH</sub> and V<sub>IL</sub> below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.4.0 V ≤ V<sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V<sub>b</sub> ≤ 4.0 V: V<sub>IH</sub> = 2.2 V, V<sub>IL</sub> = 0.8 V2.7 V ≤ V<sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V<sub>b</sub> ≤ 2.7 V: V<sub>IH</sub> = 2.0 V, V<sub>IL</sub> = 0.5 V

5. CSI00 cannot communicate at different potential. Use CSI01, CSI10, CSI11, CSI20, and CSI21 for communication at different potential.

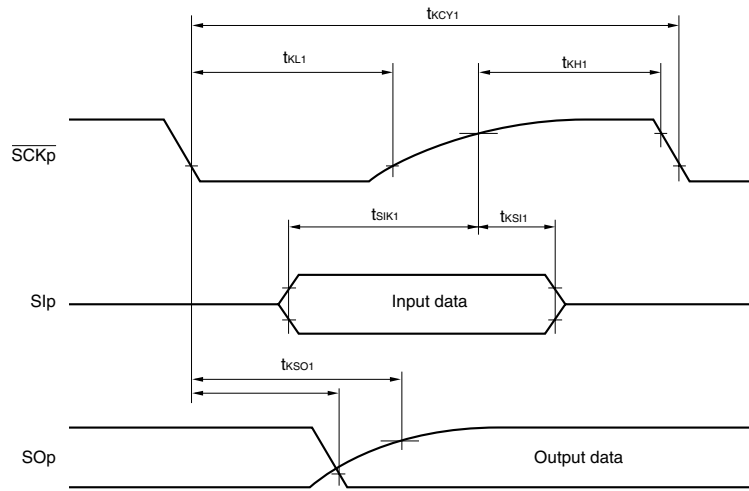
**(3) Serial interface: Serial array unit (12/18)** $(T_A = -40 \text{ to } +85^\circ\text{C}, 2.7 \text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5 \text{ V}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 \text{ V})$ **<R> (f) During communication at different potential (2.5 V, 3 V) (CSI mode) (master mode,  $\overline{\text{SCKp}}$ ... internal clock output) (2/2)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Slp setup time (to $\overline{\text{SCKp}}\downarrow$ ) <sup>Note</sup>	$t_{\text{SIK1}}$	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}, 2.7 \text{ V} \leq V_b \leq 4.0 \text{ V},$ $C_b = 30 \text{ pF}, R_b = 1.4 \text{ k}\Omega$	70			ns
		$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}, 2.3 \text{ V} \leq V_b \leq 2.7 \text{ V},$ $C_b = 30 \text{ pF}, R_b = 2.7 \text{ k}\Omega$	100			ns
Slp hold time (from $\overline{\text{SCKp}}\downarrow$ ) <sup>Note</sup>	$t_{\text{KSI1}}$	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}, 2.7 \text{ V} \leq V_b \leq 4.0 \text{ V},$ $C_b = 30 \text{ pF}, R_b = 1.4 \text{ k}\Omega$	30			ns
		$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}, 2.3 \text{ V} \leq V_b \leq 2.7 \text{ V},$ $C_b = 30 \text{ pF}, R_b = 2.7 \text{ k}\Omega$	30			ns
Delay time from $\overline{\text{SCKp}}\uparrow$ to SO <sub>p</sub> output <sup>Note</sup>	$t_{\text{KSO1}}$	$4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}, 2.7 \text{ V} \leq V_b \leq 4.0 \text{ V},$ $C_b = 30 \text{ pF}, R_b = 1.4 \text{ k}\Omega$			40	ns
		$2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}, 2.3 \text{ V} \leq V_b \leq 2.7 \text{ V},$ $C_b = 30 \text{ pF}, R_b = 2.7 \text{ k}\Omega$			40	ns

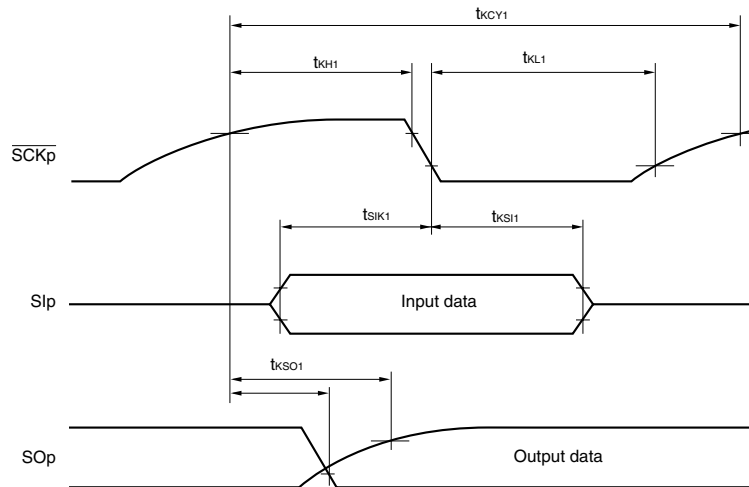
**Note** When DAP<sub>mn</sub> = 0 and CKP<sub>mn</sub> = 1, or DAP<sub>mn</sub> = 1 and CKP<sub>mn</sub> = 0.**CSI mode connection diagram (during communication at different potential)****Caution** Select the TTL input buffer for Slp and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for SO<sub>p</sub> and  $\overline{\text{SCKp}}$  by using the PIMg and POMg registers.**Remarks 1.** p: CSI number (p = 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14)**2.** m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)**3.**  $R_b[\Omega]$ : Communication line ( $\overline{\text{SCKp}}$ , SO<sub>p</sub>) pull-up resistance, $C_b[\text{F}]$ : Communication line ( $\overline{\text{SCKp}}$ , SO<sub>p</sub>) load capacitance,  $V_b[\text{V}]$ : Communication line voltage**4.**  $V_{IH}$  and  $V_{IL}$  below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode. $4.0 \text{ V} \leq V_{DD} \leq 5.5 \text{ V}, 2.7 \text{ V} \leq V_b \leq 4.0 \text{ V}: V_{IH} = 2.2 \text{ V}, V_{IL} = 0.8 \text{ V}$  $2.7 \text{ V} \leq V_{DD} < 4.0 \text{ V}, 2.3 \text{ V} \leq V_b \leq 2.7 \text{ V}: V_{IH} = 2.0 \text{ V}, V_{IL} = 0.5 \text{ V}$ **5.** CSI00 cannot communicate at different potential. Use CSI01, CSI10, CSI11, CSI20, and CSI21 for communication at different potential.

## (3) Serial interface: Serial array unit (13/18)

**CSI mode serial transfer timing (during communication at different potential)**  
**(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)**



**CSI mode serial transfer timing (during communication at different potential)**  
**(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)**



**Caution** Select the TTL input buffer for  $\text{Slp}$  and the N-ch open-drain output ( $V_{\text{DD}}$  tolerance) mode for  $\text{SOp}$  and  $\overline{\text{SCKp}}$  by using the PIMg and POMg registers.

- Remarks**
1. p: CSI number (p = 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14)
  2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
  3. CSI00 cannot communicate at different potential. Use CSI01, CSI10, CSI11, CSI20, and CSI21 for communication at different potential.

(3) Serial interface: Serial array unit (14/18)

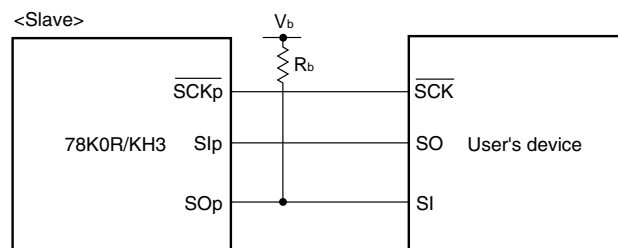
( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

<R> (g) During communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode,  $\overline{\text{SCKp}}$ ... external clock input)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCKp}}$ cycle time	$t_{\text{KCY2}}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$	$13.6\text{ MHz} < f_{\text{MCK}}$	$10/f_{\text{MCK}}$		ns
			$6.8\text{ MHz} < f_{\text{MCK}} \leq 13.6\text{ MHz}$	$8/f_{\text{MCK}}$		ns
			$f_{\text{MCK}} \leq 6.8\text{ MHz}$	$6/f_{\text{MCK}}$		ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$ , $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$	$18.5\text{ MHz} < f_{\text{MCK}}$	$16/f_{\text{MCK}}$		ns
			$14.8\text{ MHz} < f_{\text{MCK}} \leq 18.5\text{ MHz}$	$14/f_{\text{MCK}}$		ns
			$11.1\text{ MHz} < f_{\text{MCK}} \leq 14.8\text{ MHz}$	$12/f_{\text{MCK}}$		ns
			$7.4\text{ MHz} < f_{\text{MCK}} \leq 11.1\text{ MHz}$	$10/f_{\text{MCK}}$		ns
			$3.7\text{ MHz} < f_{\text{MCK}} \leq 7.4\text{ MHz}$	$8/f_{\text{MCK}}$		ns
		$f_{\text{MCK}} \leq 3.7\text{ MHz}$	$6/f_{\text{MCK}}$		ns	
$\overline{\text{SCKp}}$ high-/low-level width	$t_{\text{KH2}}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$	$f_{\text{KCY2}}/2 - 20$			ns
	$t_{\text{KL2}}$	$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$ , $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$	$f_{\text{KCY2}}/2 - 35$			ns
Slp setup time (to $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 1</sup>	$t_{\text{SIK2}}$		90			ns
Slp hold time (from $\overline{\text{SCKp}}\uparrow$ ) <sup>Note 2</sup>	$t_{\text{KSI2}}$		$1/f_{\text{MCK}} + 50$			ns
Delay time from $\overline{\text{SCKp}}\downarrow$ to SOp output <sup>Note 3</sup>	$t_{\text{KSO2}}$	$4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}$ , $2.7\text{ V} \leq V_b \leq 4.0\text{ V}$ , $C_b = 30\text{ pF}$ , $R_b = 1.4\text{ k}\Omega$			$2/f_{\text{MCK}} + 120$	ns
		$2.7\text{ V} \leq V_{DD} < 4.0\text{ V}$ , $2.3\text{ V} \leq V_b \leq 2.7\text{ V}$ , $C_b = 30\text{ pF}$ , $R_b = 2.7\text{ k}\Omega$			$2/f_{\text{MCK}} + 230$	ns

- Notes**
- When  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 0$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 1$ . The Slp setup time becomes “to  $\overline{\text{SCKp}}\downarrow$ ” when  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 1$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 0$ .
  - When  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 0$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 1$ . The Slp hold time becomes “from  $\overline{\text{SCKp}}\downarrow$ ” when  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 1$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 0$ .
  - When  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 0$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 1$ . The delay time to SOp output becomes “from  $\overline{\text{SCKp}}\uparrow$ ” when  $\text{DAPmn} = 0$  and  $\text{CKPmn} = 1$ , or  $\text{DAPmn} = 1$  and  $\text{CKPmn} = 0$ .

CSI mode connection diagram (during communication at different potential)



(Caution and Remarks are given on the next page.)

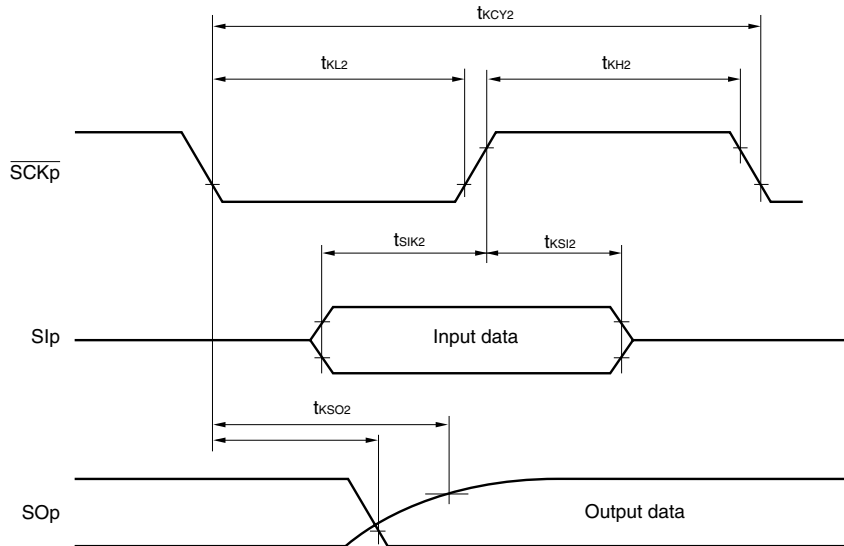
**(3) Serial interface: Serial array unit (15/18)**

**Caution** Select the TTL input buffer for  $S_{Ip}$  and  $\overline{SCKp}$  and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for  $S_{Op}$  by using the  $PIMg$  and  $POMg$  registers.

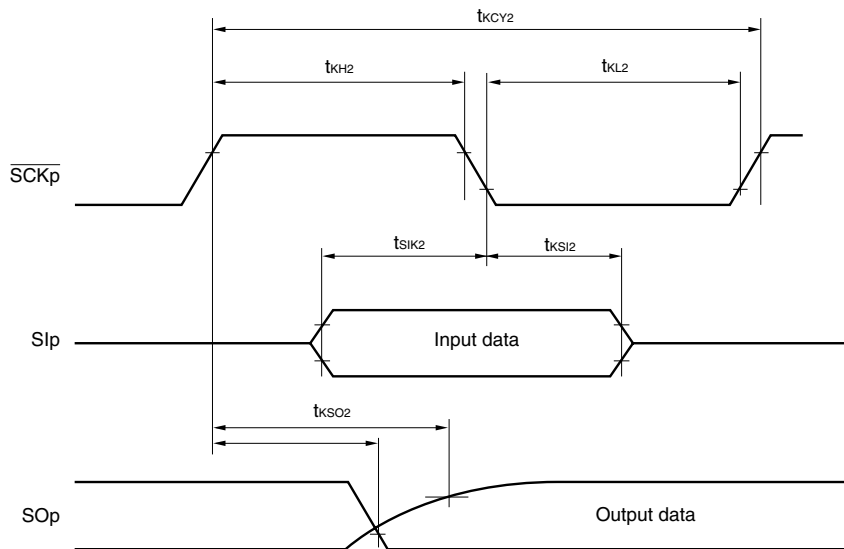
- Remarks**
1.  $p$ : CSI number ( $p = 01, 10, 11, 20, 21$ ),  $g$ : PIM and POM number ( $g = 0, 4, 9, 12, 14$ )
  2.  $R_b[\Omega]$ : Communication line ( $S_{Op}$ ) pull-up resistance,  
 $C_b[F]$ : Communication line ( $S_{Op}$ ) load capacitance,  $V_b[V]$ : Communication line voltage
  3.  $f_{MCK}$ : Serial array unit operation clock frequency  
 (Operation clock to be set by the  $CKS_{mn}$  bit of the  $SMR_{mn}$  register.  $m$ : Unit number ( $m = 0, 3$ ),  
 $n$ : Channel number ( $n = 0$  to  $2$ ))
  4.  $V_{IH}$  and  $V_{IL}$  below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.  
 $4.0\text{ V} \leq V_{DD} \leq 5.5\text{ V}, 2.7\text{ V} \leq V_b \leq 4.0\text{ V}$ :  $V_{IH} = 2.2\text{ V}, V_{IL} = 0.8\text{ V}$   
 $2.7\text{ V} \leq V_{DD} \leq 4.0\text{ V}, 2.3\text{ V} \leq V_b \leq 2.7\text{ V}$ :  $V_{IH} = 2.0\text{ V}, V_{IL} = 0.5\text{ V}$
  5.  $CSI_{00}$  cannot communicate at different potential. Use  $CSI_{01}, CSI_{10}, CSI_{11}, CSI_{20}$ , and  $CSI_{21}$  for communication at different potential.

## (3) Serial interface: Serial array unit (16/18)

CSI mode serial transfer timing (during communication at different potential)  
 (When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1)



CSI mode serial transfer timing (during communication at different potential)  
 (When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0)



**Caution** Select the TTL input buffer for  $\text{Slp}$  and  $\overline{\text{SCKp}}$  and the N-ch open-drain output ( $V_{\text{DD}}$  tolerance) mode for  $\text{SOp}$  by using the PIMg and POMg registers.

- Remarks**
1. p: CSI number (p = 01, 10, 11, 20, 21), g: PIM and POM number (g = 0, 4, 9, 12, 14)
  2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
  3. CSI00 cannot communicate at different potential. Use CSI01, CSI10, CSI11, CSI20, and CSI21 for communication at different potential.

## (3) Serial interface: Serial array unit (17/18)

(T<sub>A</sub> = -40 to +85°C, 2.7 V ≤ V<sub>DD</sub> = EV<sub>DD0</sub> = EV<sub>DD1</sub> ≤ 5.5 V, V<sub>SS</sub> = EV<sub>SS0</sub> = EV<sub>SS1</sub> = AV<sub>SS</sub> = 0 V)(h) During communication at different potential (2.5 V, 3 V) (simplified I<sup>2</sup>C mode)

&lt;R&gt;

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f <sub>SCL</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 1.4 kΩ		400 <sup>Note</sup>	kHz
		2.7 V ≤ V <sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ		400 <sup>Note</sup>	kHz
Hold time when SCLr = "L"	t <sub>LOW</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 1.4 kΩ	1065		ns
		2.7 V ≤ V <sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	1065		ns
Hold time when SCLr = "H"	t <sub>HIGH</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 1.4 kΩ	445		ns
		2.7 V ≤ V <sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	445		ns
Data setup time (reception)	t <sub>SU:DAT</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 1.4 kΩ	1/f <sub>MCK</sub> + 190		ns
		2.7 V ≤ V <sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	1/f <sub>MCK</sub> + 190		ns
Data hold time (transmission)	t <sub>HD:DAT</sub>	4.0 V ≤ V <sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V <sub>b</sub> ≤ 4.0 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 1.4 kΩ	0	160	ns
		2.7 V ≤ V <sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V <sub>b</sub> ≤ 2.7 V, C <sub>b</sub> = 100 pF, R <sub>b</sub> = 2.7 kΩ	0	160	ns

&lt;R&gt;

**Note** The value must also be f<sub>MCK</sub>/4 or less.

**Caution** Select the TTL input buffer and the N-ch open-drain output (V<sub>DD</sub> tolerance) mode for SDAr and the N-ch open-drain output (V<sub>DD</sub> tolerance) mode for SCLr by using the PIMg and POMg registers.

**Remarks 1.** R<sub>b</sub>[Ω]: Communication line (SDAr, SCLr) pull-up resistance,

C<sub>b</sub>[F]: Communication line (SDAr, SCLr) load capacitance, V<sub>b</sub>[V]: Communication line voltage

**2.** r: IIC number (r = 10, 11, 20, 21), g: PIM, POM number (g = 0, 9, 12, 14)

**3.** f<sub>MCK</sub>: Serial array unit operation clock frequency

(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1),

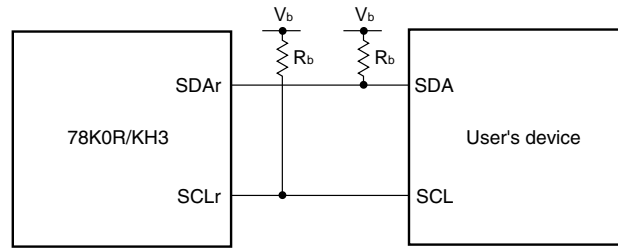
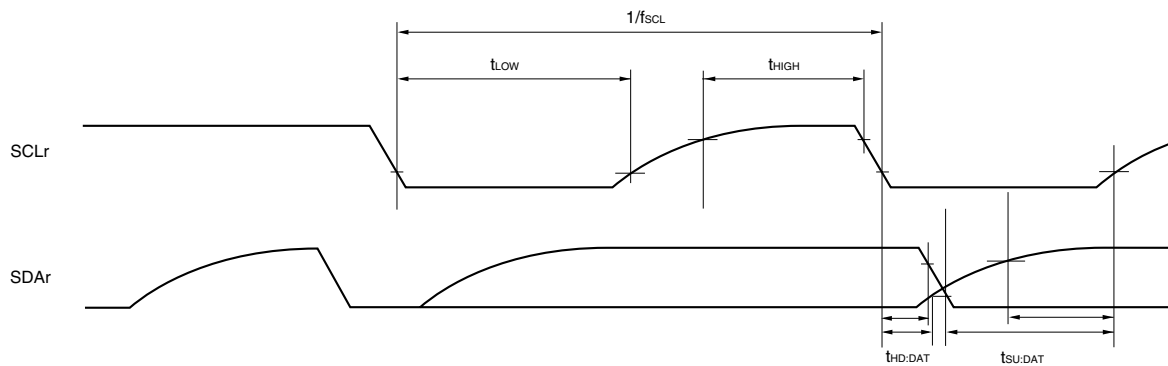
n: Channel number (n = 0, 3), mn = 02, 03, 10, 11)

**4.** V<sub>IH</sub> and V<sub>IL</sub> below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in simplified I<sup>2</sup>C mode.

4.0 V ≤ V<sub>DD</sub> ≤ 5.5 V, 2.7 V ≤ V<sub>b</sub> ≤ 4.0 V: V<sub>IH</sub> = 2.2 V, V<sub>IL</sub> = 0.8 V

2.7 V ≤ V<sub>DD</sub> ≤ 4.0 V, 2.3 V ≤ V<sub>b</sub> ≤ 2.7 V: V<sub>IH</sub> = 2.0 V, V<sub>IL</sub> = 0.5 V

## (3) Serial interface: Serial array unit (18/18)

Simplified I<sup>2</sup>C mode connection diagram (during communication at different potential)Simplified I<sup>2</sup>C mode serial transfer timing (during communication at different potential)

**Caution** Select the TTL input buffer and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for SDAr and the N-ch open-drain output ( $V_{DD}$  tolerance) mode for SCLr by using the PIMg and POMg registers.

- Remarks**
- $R_b[\Omega]$ : Communication line (SDAr, SCLr) pull-up resistance,  $V_b[V]$ : Communication line voltage
  - r: IIC number (r = 10, 11, 20, 21), g: PIM and POM number (g = 0, 9, 12, 14)



(4) Serial interface: IIC0

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

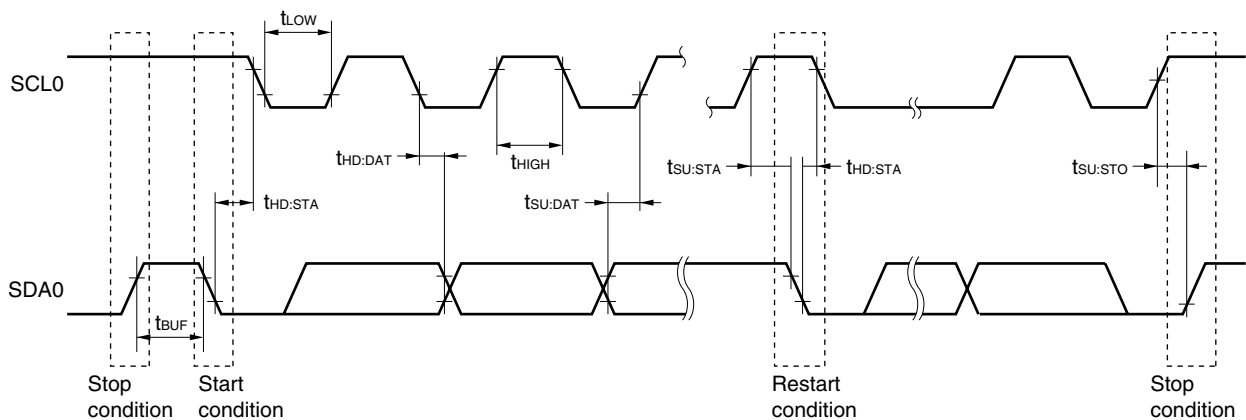
(a) IIC0

Parameter	Symbol	Conditions	Standard Mode		Fast Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCL0 clock frequency	$f_{SCL}$	$6.7\text{ MHz} \leq f_{CLK}$	0	100	0	400	kHz
		$4.0\text{ MHz} \leq f_{CLK} < 6.7\text{ MHz}$	0	100	0	340	kHz
		$3.2\text{ MHz} \leq f_{CLK} < 4.0\text{ MHz}$	0	100	–	–	kHz
		$2.0\text{ MHz} \leq f_{CLK} < 3.2\text{ MHz}$	0	85	–	–	kHz
Setup time of restart condition <sup>Note 1</sup>	$t_{SU:STA}$		4.7		0.6		$\mu\text{s}$
Hold time	$t_{HD:STA}$		4.0		0.6		$\mu\text{s}$
Hold time when SCL0 = "L"	$t_{LOW}$		4.7		1.3		$\mu\text{s}$
Hold time when SCL0 = "H"	$t_{HIGH}$		4.0		0.6		$\mu\text{s}$
Data setup time (reception)	$t_{SU:DAT}$		250		100		ns
Data hold time (transmission) <sup>Note 2</sup>	$t_{HD:DAT}$	CL00 = 1 and CL01 = 1	0	3.45 <sup>Note 3</sup>	0	0.9 <sup>Note 4</sup>	$\mu\text{s}$
				5.50 <sup>Note 5</sup>		1.5 <sup>Note 6</sup>	$\mu\text{s}$
		CL00 = 0 and CL01 = 0, or CL00 = 1 and CL01 = 0	0	3.45	0	0.9 <sup>Note 7</sup>	$\mu\text{s}$
						0.95 <sup>Note 8</sup>	$\mu\text{s}$
CL00 = 0 and CL01 = 1	0	3.45	0	0.9	$\mu\text{s}$		
Setup time of stop condition	$t_{SU:STO}$		4.0		0.6		$\mu\text{s}$
Bus-free time	$t_{BUF}$		4.7		1.3		$\mu\text{s}$

- Notes**
1. The first clock pulse is generated after this period when the start/restart condition is detected.
  2. The maximum value (MAX.) of  $t_{HD:DAT}$  is during normal transfer and a wait state is inserted in the  $\overline{\text{ACK}}$  (acknowledge) timing.
  3. When  $3.2\text{ MHz} \leq f_{CLK} \leq 4.19\text{ MHz}$ .
  4. When  $6.7\text{ MHz} \leq f_{CLK} \leq 8.38\text{ MHz}$ .
  5. When  $2.0\text{ MHz} \leq f_{CLK} < 3.2\text{ MHz}$ . At this time, use the SCL0 clock within 85 kHz.
  6. When  $4.0\text{ MHz} \leq f_{CLK} < 6.7\text{ MHz}$ . At this time, use the SCL0 clock within 340 kHz.
  7. When  $8.0\text{ MHz} \leq f_{CLK} \leq 16.76\text{ MHz}$ .
  8. When  $7.6\text{ MHz} \leq f_{CLK} < 8.0\text{ MHz}$ .

**Remark** CL00, CL01, DFC0: Bits 0, 1, and 2 of the IIC clock select register 0 (IICCL0)

IIC0 serial transfer timing



**(5) Serial interface: On-chip debug (UART)****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )****(a) On-chip debug (UART)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate			$f_{CLK}/2^{12}$		$f_{CLK}/6$	bps
		Flash memory programming mode			2.66	Mbps
TOOL1 output frequency	$f_{TOOL1}$	$2.7\text{ V} \leq V_{DD} \leq 5.5\text{ V}$			10	MHz
		$1.8\text{ V} \leq V_{DD} < 2.7\text{ V}$			2.5	MHz

**A/D Converter Characteristics (1/2)****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.3\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $2.3\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	$R_{ES}$				10	bit
Overall error <sup>Notes 1, 2</sup>	$AINL$	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$			$\pm 0.4$	%FSR
		$2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$			$\pm 0.5$	%FSR
		$2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$			$\pm 0.7$	%FSR
Conversion time	$t_{CONV}$	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$	6.1		66.6	$\mu\text{s}$
		$2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$	12.2		66.6	$\mu\text{s}$
		$2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$	27		66.6	$\mu\text{s}$
Zero-scale error <sup>Notes 1, 2</sup>	$EZS$	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$			$\pm 0.4$	%FSR
		$2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$			$\pm 0.5$	%FSR
		$2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$			$\pm 0.5$	%FSR
Full-scale error <sup>Notes 1, 2</sup>	$EFS$	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$			$\pm 0.4$	%FSR
		$2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$			$\pm 0.5$	%FSR
		$2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$			$\pm 0.5$	%FSR
Integral linearity error <sup>Note 1</sup>	$ILE$	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$			$\pm 2.5$	LSB
		$2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$			$\pm 3.5$	LSB
		$2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$			$\pm 3.5$	LSB
Differential linearity error <sup>Note 1</sup>	$DLE$	$4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$			$\pm 1.5$	LSB
		$2.7\text{ V} \leq AV_{REF0} < 4.0\text{ V}$			$\pm 1.5$	LSB
		$2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$			$\pm 1.5$	LSB
Analog input voltage	$V_{AIN}$	$2.3\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$	$AV_{SS}$		$AV_{REF0}$	V

**Notes 1.** Excludes quantization error ( $\pm 1/2$  LSB).**2.** This value is indicated as a ratio (%FSR) to the full-scale value.

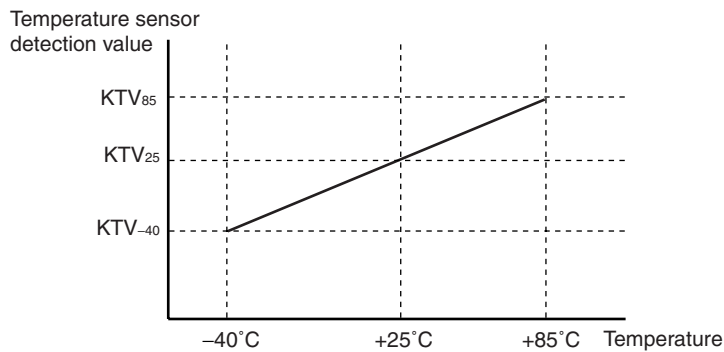
**Temperature Sensor (Expanded-Specification Products)**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $2.7\text{ V} \leq AV_{REF0} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = 0\text{ V}$ )

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Augmentation factor per 10°C	TC		1	3.5	15	/10°C
Temperature sensor detection value	KTV <sub>-40</sub>	$T_A = -40^\circ\text{C}$	30	80	130	–
	KTV <sub>25</sub>	$T_A = +25^\circ\text{C}$	65	101	140	–
	KTV <sub>85</sub>	$T_A = +85^\circ\text{C}$	100	122	150	–

**Remark** The temperature sensor detection value is obtained by using the following expression.

$$\text{Temperature sensor detection value} = \frac{\text{A/D conversion value with sensor that depends on temperature}}{\text{A/D conversion value with sensor that does not depend on temperature}} \times 256 \cong \frac{TC}{10} \left( \text{Temperature during sensor operation} - \text{Low reference temperature} \right) + \text{Temperature sensor detection value at a low reference temperature}$$



**D/A Converter Characteristics**

( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $1.8\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $1.8\text{ V} \leq AV_{REF1} \leq V_{DD}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0\text{ V}$ )

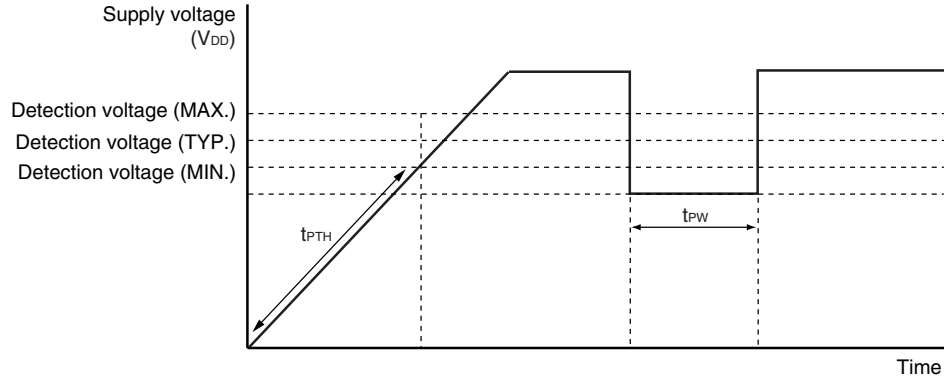
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Resolution	RES				8	bit	
Overall error	AINL	$R_{load} = 2\text{ M}\Omega$			$\pm 1.2$	%FSR	
		$R_{load} = 4\text{ M}\Omega$			$\pm 0.8$	%FSR	
		$R_{load} = 10\text{ M}\Omega$			$\pm 0.6$	%FSR	
Settling time	$t_{SET}$	$C_{load} = 20\text{ pF}$	$4.0\text{ V} \leq AV_{REF1} \leq 5.5\text{ V}$			3	$\mu\text{s}$
			$2.7\text{ V} \leq AV_{REF1} < 4.0\text{ V}$			3	$\mu\text{s}$
			$1.8\text{ V} \leq AV_{REF1} < 2.7\text{ V}$			6	$\mu\text{s}$
D/A output resistance value	$R_o$	Per D/A converter channel		6.4		$\text{k}\Omega$	

**Remark** When the D/A converter is in normal mode, D/A conversion is started after one  $f_{CLK}$  clock has elapsed since the DACSn register was written. The output level is determined when the settling time has elapsed after the D/A conversion was started.

**POC Circuit Characteristics ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	$V_{POC0}$		1.5	1.59	1.68	V
Power supply voltage rise inclination	$t_{PTH}$	Change inclination of $V_{DD}$ : $0\text{ V} \rightarrow V_{POC0}$	0.5			V/ms
Minimum pulse width	$t_{PW}$	When the voltage drops	200			$\mu\text{s}$
Detection delay time					200	$\mu\text{s}$

**POC Circuit Timing**



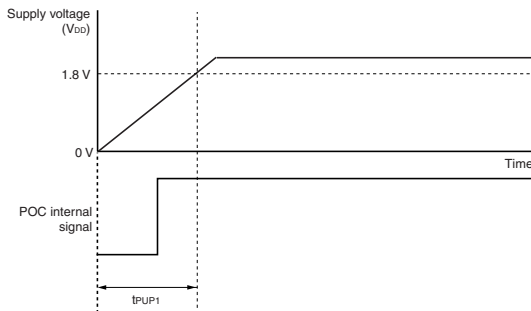
**Supply Voltage Rise Time ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{SS} = 0\text{ V}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 1.8 V ( $V_{DD}(\text{MIN.})$ ) <sup>Note</sup> ( $V_{DD}$ : $0\text{ V} \rightarrow 1.8\text{ V}$ )	$t_{PUP1}$	LVI default start function stopped is set ( $\overline{\text{LVIOFF}}$ (option byte) = 1), when $\overline{\text{RESET}}$ input is not used			3.6	ms
Maximum time to rise to 1.8 V ( $V_{DD}(\text{MIN.})$ ) <sup>Note</sup> (releasing $\overline{\text{RESET}}$ input $\rightarrow V_{DD}$ : 1.8 V)	$t_{PUP2}$	LVI default start function stopped is set ( $\overline{\text{LVIOFF}}$ (option byte) = 1), when $\overline{\text{RESET}}$ input is used			1.88	ms

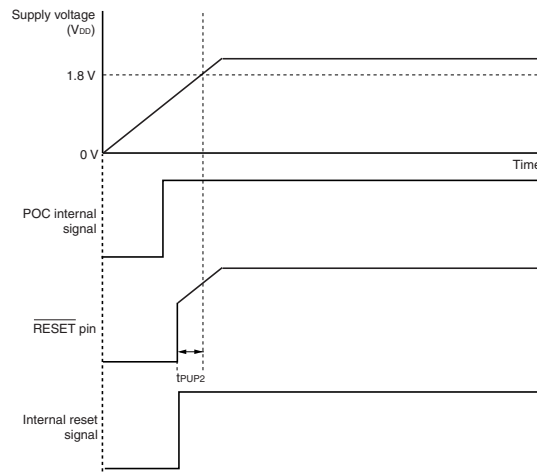
**Note** Make sure to raise the power supply in a shorter time than this.

**Supply Voltage Rise Time Timing**

- When  $\overline{\text{RESET}}$  pin input is not used



- When  $\overline{\text{RESET}}$  pin input is used (when external reset is released by the  $\overline{\text{RESET}}$  pin, after POC has been released)



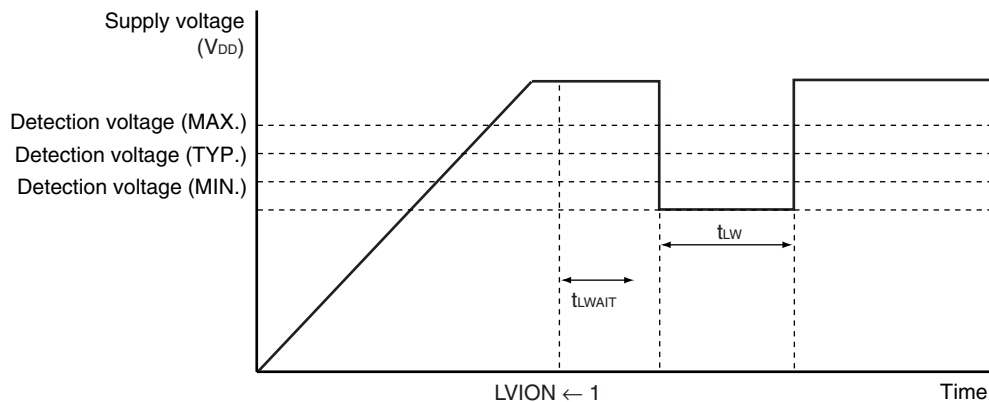
**LVI Circuit Characteristics** ( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $V_{POC} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5$  V,  $V_{SS} = EV_{SS0} = EV_{SS1} = 0$  V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Detection voltage	Supply voltage level	$V_{LVI0}$		4.12	4.22	4.32	V
		$V_{LVI1}$		3.97	4.07	4.17	V
		$V_{LVI2}$		3.82	3.92	4.02	V
		$V_{LVI3}$		3.66	3.76	3.86	V
		$V_{LVI4}$		3.51	3.61	3.71	V
		$V_{LVI5}$		3.35	3.45	3.55	V
		$V_{LVI6}$		3.20	3.30	3.40	V
		$V_{LVI7}$		3.05	3.15	3.25	V
		$V_{LVI8}$		2.89	2.99	3.09	V
		$V_{LVI9}$		2.74	2.84	2.94	V
		$V_{LVI10}$		2.58	2.68	2.78	V
		$V_{LVI11}$		2.43	2.53	2.63	V
		$V_{LVI12}$		2.28	2.38	2.48	V
		$V_{LVI13}$		2.12	2.22	2.32	V
		$V_{LVI14}$		1.97	2.07	2.17	V
$V_{LVI15}$		1.81	1.91	2.01	V		
External input pin <sup>Note 1</sup>	$V_{EXLVI}$	$EXLVI < V_{DD}$ , $1.8\text{ V} \leq V_{DD} \leq 5.5\text{ V}$	1.11	1.21	1.31	V	
Power supply voltage on power application	$V_{PULVI}$	When LVI default start function enabled is set	1.87	2.07	2.27	V	
Minimum pulse width	$t_{LW}$		200			$\mu\text{s}$	
Detection delay time					200	$\mu\text{s}$	
Operation stabilization wait time <sup>Note 2</sup>	$t_{LWAIT}$				10	$\mu\text{s}$	

**Notes** 1. The EXLVI/P120/INTP0 pin is used.

2. Time required from setting bit 7 (LVION) of the low-voltage detection register (LVIM) to 1 to operation stabilization

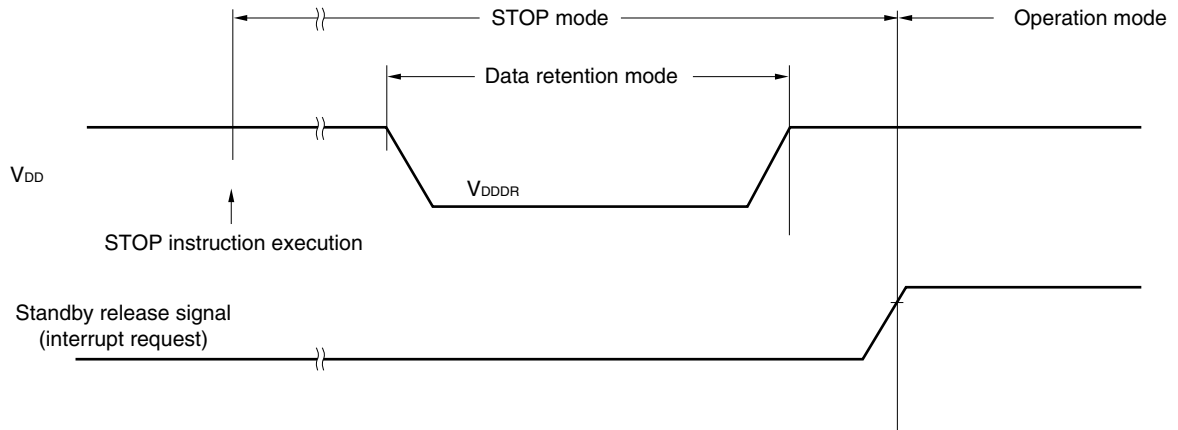
**Remark**  $V_{LVI(n-1)} > V_{LVI n}$ :  $n = 1$  to  $15$

**LVI Circuit Timing**

**Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics ( $T_A = -40$  to  $+85^\circ\text{C}$ )**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	$V_{DDDR}$		1.5 <sup>Note</sup>		5.5	V

**Note** The value depends on the POC detection voltage. When the voltage drops, the data is retained until a POC reset is effected, but data is not retained when a POC reset is effected.



**Flash Memory Programming Characteristics****( $T_A = -40$  to  $+85^\circ\text{C}$ ,  $2.7\text{ V} \leq V_{DD} = EV_{DD0} = EV_{DD1} \leq 5.5\text{ V}$ ,  $V_{SS} = EV_{SS0} = EV_{SS1} = 0\text{ V}$ )**

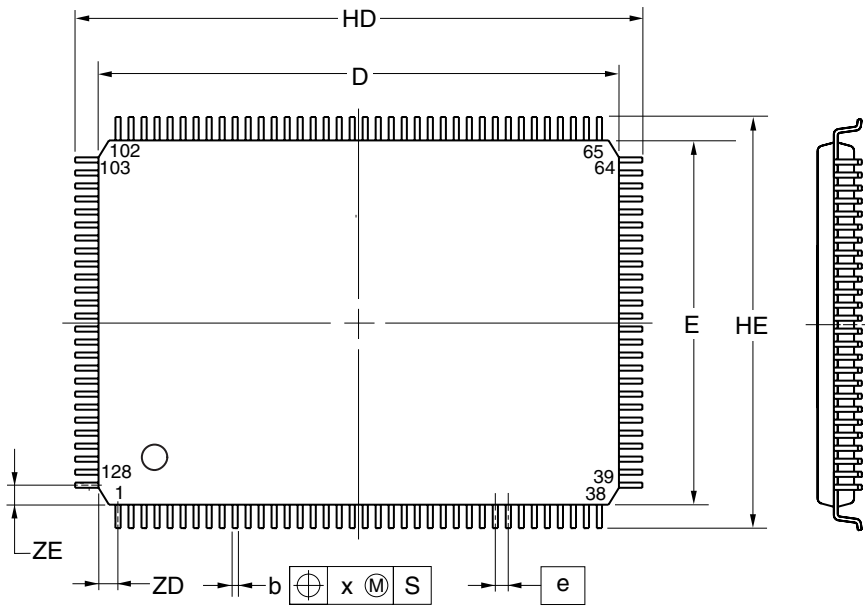
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$V_{DD}$ supply current	$I_{DD}$	TYP. = 10 MHz, MAX. = 20 MHz			4.5	15	mA
CPU/peripheral hardware clock frequency	$f_{CLK}$			2		20	MHz
Number of rewrites (number of deletes per block)	$C_{WRT}$	Used for updating programs When using flash memory programmer and NEC Electronics self programming library	Retained for 15 years	100			Times
		Used for updating data When using NEC Electronics EEPROM emulation library (usable ROM size: 6 KB of 3 consecutive blocks)	Retained for 5 years	10,000			Times

**Remark** When updating data multiple times, use the flash memory as one for updating data.

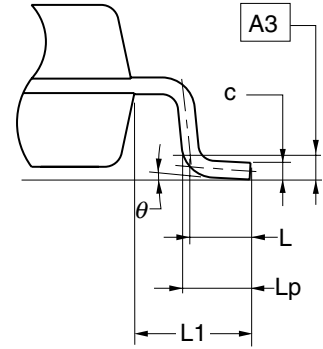


CHAPTER 31 PACKAGE DRAWINGS

128-PIN PLASTIC LQFP (FINE PITCH) (14x20)

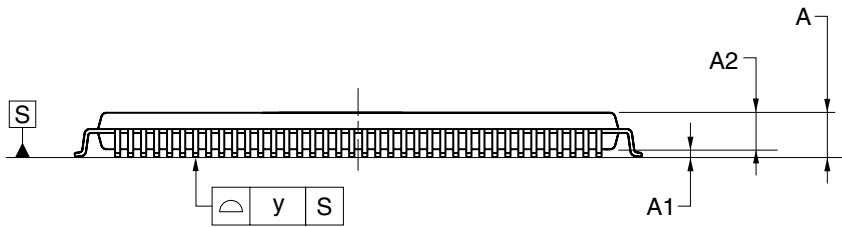


detail of lead end



(UNIT:mm)

ITEM	DIMENSIONS
D	20.00±0.20
E	14.00±0.20
HD	22.00±0.20
HE	16.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.20 <sup>+0.07</sup> <sub>-0.03</sub>
c	0.125 <sup>+0.075</sup> <sub>-0.025</sub>
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	3° <sup>+5°</sup> <sub>-3°</sub>
e	0.50
x	0.08
y	0.08
ZD	0.75
ZE	0.75



NOTE

Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

P128GF-50-GAT

© NEC Electronics Corporation 2006

## CHAPTER 32 RECOMMENDED SOLDERING CONDITIONS

These products should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (<http://www.necel.com/pkg/en/mount/index.html>)

**Caution** For soldering methods and conditions other than those recommended below, contact an NEC Electronics sales representative.

**Table 32-1. Surface Mounting Type Soldering Conditions**

• **128-pin plastic LQFP (14 × 20)**

$\mu$ PD78F1174GF-GAT-AX,  $\mu$ PD78F1174AGF-GAT-AX,  $\mu$ PD78F1174AGF(A)-GAT-AX,  
 $\mu$ PD78F1175GF-GAT-AX,  $\mu$ PD78F1175AGF-GAT-AX,  $\mu$ PD78F1175AGF(A)-GAT-AX,  
 $\mu$ PD78F1176GF-GAT-AX,  $\mu$ PD78F1176AGF-GAT-AX,  $\mu$ PD78F1176AGF(A)-GAT-AX,  
 $\mu$ PD78F1177GF-GAT-AX,  $\mu$ PD78F1177AGF-GAT-AX,  $\mu$ PD78F1177AGF(A)-GAT-AX,  
 $\mu$ PD78F1178GF-GAT-AX,  $\mu$ PD78F1178AGF-GAT-AX,  $\mu$ PD78F1178AGF(A)-GAT-AX

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: 3 times or less, Exposure limit: 7 days <sup>Note</sup> (after that, prebake at 125°C for 10 to 72 hours)	IR60-107-3
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	-

**Note** After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

**Caution** Do not use different soldering methods together (except for partial heating).

## APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the 78K0R/KH3. Figure A-1 shows the development tool configuration.

- **Support for PC98-NX series**

Unless otherwise specified, products supported by IBM PC/AT™ compatibles are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT compatibles.

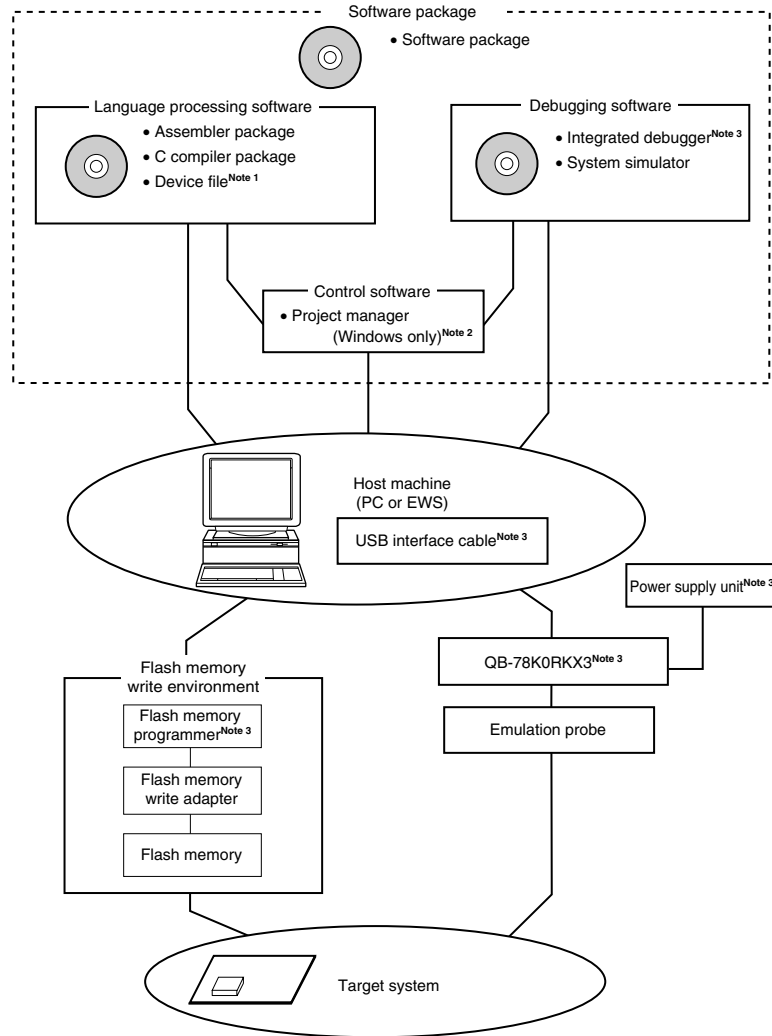
- **Windows™**

Unless otherwise specified, "Windows" means the following OSs.

- Windows 98
- Windows NT™
- Windows 2000
- Windows XP

Figure A-1. Development Tool Configuration (1/2)

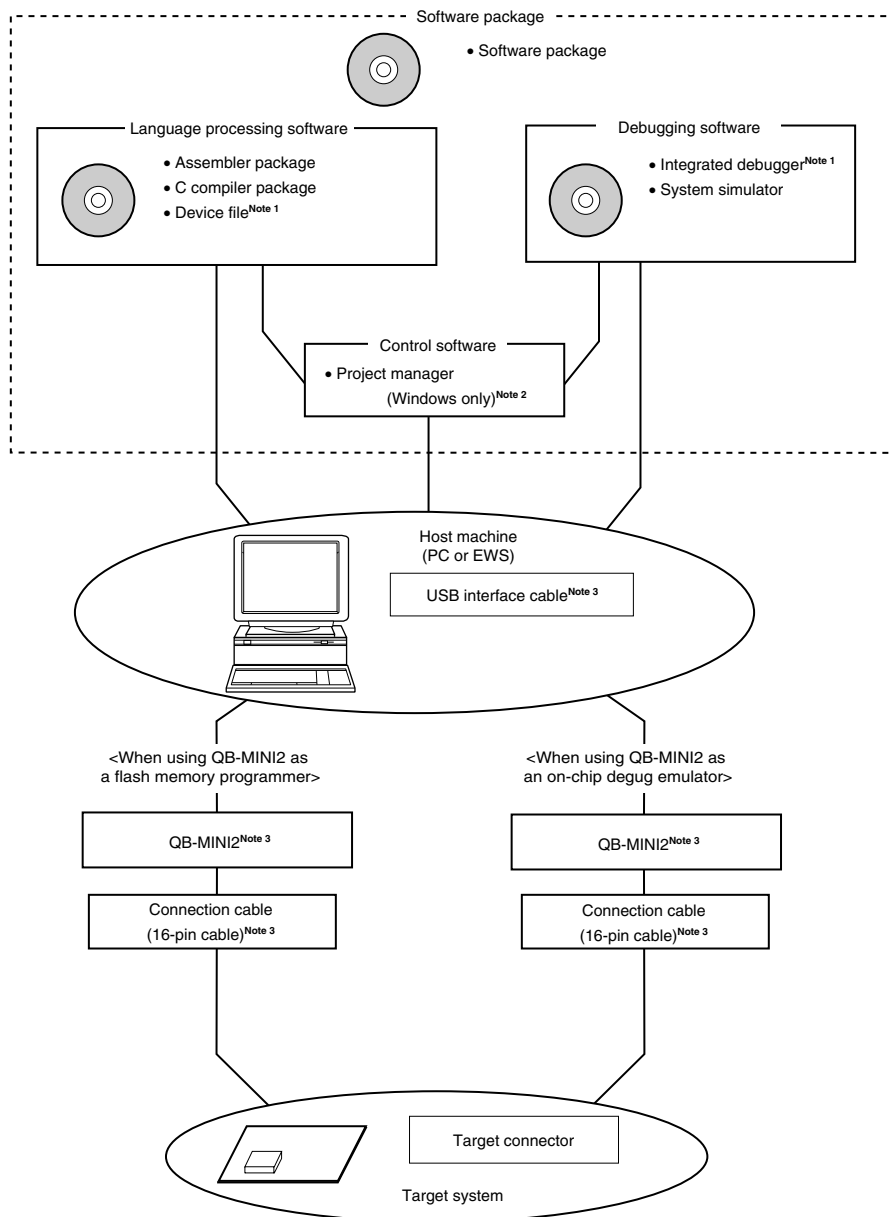
(1) When using the in-circuit emulator QB-78K0RKX3



- Notes**
1. Download the device file for 78K0R/KH3 (DF781188) from the download site for development tools (<http://www.necel.com/micro/ods/eng/index.html>).
  2. The project manager PM+ is included in the assembler package. The PM+ is only used for Windows.
  3. In-circuit emulator QB-78K0RKX3 is supplied with integrated debugger ID78K0R-QB, on-chip debug emulator with programming function QB-MINI2, power supply unit, and USB interface cable. Any other products are sold separately.

Figure A-1. Development Tool Configuration (2/2)

## (2) When using the on-chip debug emulator with programming function QB-MINI2



- Notes**
1. Download the device file for 78K0R/KH3 (DF781188) and the integrated debugger (ID78K0R-QB) from the download site for development tools (<http://www.necel.com/micro/ods/eng/index.html>).
  2. The project manager PM+ is included in the assembler package.  
The PM+ is only used for Windows.
  3. On-chip debug emulator QB-MINI2 is supplied with USB interface cable, connection cables (10-pin cable and 16-pin cable), and 78K0-OCD board. Any other products are sold separately. In addition, download the software for operating the QB-MINI2 from the download site for MINICUBE2 (<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>).

### A.1 Software Package

SP78K0R 78K0R Series software package	Development tools (software) common to the 78K0R microcontrollers are combined in this package.
	Part number: $\mu$ SxxxxSP78K0R

**Remark** xxxx in the part number differs depending on the host machine and OS used.

$\mu$ SxxxxSP78K0R

xxxx	Host Machine	OS	Supply Medium
AB17	PC-9800 series,	Windows (Japanese version)	CD-ROM
BB17	IBM PC/AT compatibles	Windows (English version)	

### A.2 Language Processing Software

RA78K0R Assembler package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. This assembler is also provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with a device file (DF781188) <b>&lt;Precaution when using RA78K0R in PC environment&gt;</b> This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: $\mu$ SxxxxRA78K0R
CC78K0R C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an assembler package and device file (both sold separately). <b>&lt;Precaution when using CC78K0R in PC environment&gt;</b> This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: $\mu$ SxxxxCC78K0R
DF781188 <sup>Note</sup> Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0R, CC78K0R, SM+ for 78K0R, and ID78K0R-QB) (all sold separately). The corresponding OS and host machine differ depending on the tool to be used.
	Part number: $\mu$ SxxxxDF781188

**Note** The DF781188 can be used in common with the RA78K0R, CC78K0R, SM+ for 78K0R, and ID78K0R-QB. Download the DF781188 from the download site for development tools (<http://www.necel.com/micro/ods/eng/index.html>).

**Remark** xxxx in the part number differs depending on the host machine and OS used.

μSxxxxRA78K0R

μSxxxxCC78K0R

xxxx	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

μSxxxxDF781188

xxxx	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	

### A.3 Control Software

PM+ Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from the project manager. <b>&lt;Caution&gt;</b> The project manager is included in the assembler package (RA78K0R). It can only be used in Windows.
------------------------	--

## A.4 Flash Memory Programming Tools

### A.4.1 When using flash memory programmers PG-FP5, FL-PR5, PG-FP4 and FL-PR4

PG-FP5, FL-PR5, PG-FP4, FL-PR4 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
FA-78F1178GF-GAT-RX (RoHS supported), Flash memory programming adapter	Flash memory programming adapter used connected to the flash memory programmer for use.

**Remark** The FL-PR4, FL-PR5, FA-78F1178GF-GAT-RX are a product of Naito Densai Machida Mfg. Co., Ltd.

### A.4.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This is a flash memory programmer dedicated to microcontrollers with on-chip flash memory. It is available also as on-chip debug emulator which serves to debug hardware and software when developing application systems using the 78K0R. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. To use 78K0R/KH3, use USB interface cable and 16-pin connection cable.
---	---

**Remark** Download the software for operating the QB-MINI2 from the download site for MINICUBE2 (<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>).



## A.5 Debugging Tools (Hardware)

### A.5.1 When using in-circuit emulator QB-78K0RKX3

QB-78K0KX3 In-circuit emulator	This in-circuit emulator serves to debug hardware and software when developing application systems using the 78K0R/Kx3. It supports to the integrated debugger (ID78K0R-QB). This emulator should be used in combination with a power supply unit and emulation probe, and the USB is used to connect this emulator to the host machine.
QB-144-CA-01 Check pin adapter	This check pin adapter is used in waveform monitoring using the oscilloscope, etc.
QB-144-EP-02S Emulation probe	This emulation probe is flexible type and used to connect the in-circuit emulator and target system.
QB-128GF-EA-01T Exchange adapter	This exchange adapter is used to perform pin conversion from the in-circuit emulator to target connector.
QB-128GF-YS-01T Space adapter	This space adapter is used to adjust the height between the target system and in-circuit emulator.
QB-128GF-YQ-01T YQ connector	This YQ connector is used to connect the target connector and exchange adapter.
QB-128GF-HQ-01T Mount adapter	This mount adapter is used to mount the target device with socket.
QB-128GF-NQ-01T Target connector	This target connector is used to mount on the target system.

- Remarks**
1. The QB-78K0RKX3 is supplied with a power supply unit and USB interface cable. As control software, integrated debugger ID78K0R-QB and on-chip debug emulator with programming function QB-MINI2 are supplied.
  2. The packed contents differ depending on the part number, as follows.

Packed Contents Part Number	In-Circuit Emulator	Emulation Probe	Exchange Adapter	YQ Connector	Target Connector
QB-78K0RKX3-ZZZ	QB-78K0RKX3	None			
QB-78K0RKX3-T128GF		QB-144-EP-02S	QB-128GF-EA-01T	QB-128GF-YQ-01T	QB-128GF-NQ-01T

**A.5.2 When using on-chip debug emulator with programming function QB-MINI2**

<p>QB-MINI2 On-chip debug emulator with programming function</p>	<p>This on-chip debug emulator serves to debug hardware and software when developing application systems using the 78K0R microcontrollers. It is available also as flash memory programmer dedicated to microcontrollers with on-chip flash memory. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. To use 78K0R/KH3, use USB interface cable and 16-pin connection cable.</p>
--	---

**Remark** Download the software for operating the QB-MINI2 from the download site for MINICUBE2 (<http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html>).

**A.6 Debugging Tools (Software)**

<p>SM+ for 78K0R System simulator</p>	<p>SM+ for 78K0R is Windows-based software. It is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. Use of SM+ for 78K0R allows the execution of application logical testing and performance testing on an independent basis from hardware development, thereby providing higher development efficiency and software quality. SM+ for 78K0R should be used in combination with the device file (DF781188) .</p>
<p>Part number: <math>\mu</math>SxxxxSM781000</p>	
<p>ID78K0R-QB Integrated debugger</p>	<p>This debugger supports the in-circuit emulators for the 78K0R microcontrollers. The ID78K0R-QB is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with the device file.</p>
<p>Part number: <math>\mu</math>SxxxxID78K0R-QB</p>	

**Remark** xxxx in the part number differs depending on the host machine and OS used.

$\mu$ SxxxxSM781000

$\mu$ SxxxxID78K0R-QB

xxxx	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

## APPENDIX B LIST OF CAUTIONS

This appendix lists the cautions described in this document.

“Classification (hard/soft)” in the table is as follows.

Hard: Cautions for microcontroller internal/external hardware

Soft: Cautions for software such as register settings or programs

(1/35)

Chapter	Classification	Function	Details of Function	Caution	Page	
Chapter 1	Hard	Outline	AV <sub>SS</sub> , EV <sub>SS0</sub> , EV <sub>SS1</sub> , V <sub>SS</sub>	Make AV <sub>SS</sub> , EV <sub>SS0</sub> , and EV <sub>SS1</sub> the same potential as V <sub>SS</sub> .	p.22 <input type="checkbox"/>	
			EV <sub>DD0</sub> , EV <sub>DD1</sub> , V <sub>DD</sub>	Make EV <sub>DD0</sub> and EV <sub>DD1</sub> the same potential as V <sub>DD</sub> .	p.22 <input type="checkbox"/>	
			REGC	Connect the REGC pin to V <sub>SS</sub> via a capacitor (0.47 to 1 μF).	p.22 <input type="checkbox"/>	
			P20/ANI0 to P27/ANI7, P150/ANI8 to P157/ANI15	P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 are set as analog inputs in the order of P157/ANI15, ..., P150/ANI8, P27/ANI7, ..., P20/ANI0 by the A/D port configuration register (ADPC). When using P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 as analog inputs, start designing from P157/ANI15 (see 11.3 (6) A/D port configuration register (ADPC) for details).	p.22 <input type="checkbox"/>	
Chapter 2	Soft	Pin functions	P02/SO10/TxD1, P04/SCK10/ SCL10	To use P02/SO10/TxD1 and P04/SCK10/SCL10 as general-purpose ports, set serial communication operation setting register 02 (SCR02) to the default status (0087H). In addition, clear port output mode register 0 (POM0) to 00H.	p.37 <input type="checkbox"/>	
			P10/SCK00/EX24, P12/SO00/TxD0/ EX26	To use P10/SCK00/EX24 and P12/SO00/TxD0/EX26 as general-purpose ports, set serial communication operation setting register 00 (SCR00) to the default status (0087H).	p.38 <input type="checkbox"/>	
			RTCCL, RTCDIV	Do not enable outputting RTCCL and RTCDIV at the same time.	p.38 <input type="checkbox"/>	
	Hard			ANI0/P20 to ANI7/P27	ANI0/P20 to ANI7/P27 are set in the digital input (general-purpose port) mode after release of reset.	p.38 <input type="checkbox"/>
				P40/TOOL0	The function of the P40/TOOL0 pin varies as described in (a) to (c) below. In the case of (b) or (c), make the specified connection. (a) In normal operation mode and when on-chip debugging is disabled (OCDENSET = 0) by an option byte (000C3H) => Use this pin as a port pin (P40). (b) In normal operation mode and when on-chip debugging is enabled (OCDENSET = 1) by an option byte (000C3H) => Connect this pin to EV <sub>DD0</sub> or EV <sub>DD1</sub> via an external resistor, and always input a high level to the pin before reset release. (c) When on-chip debug function is used, or in write mode of flash memory programmer => Use this pin as TOOL0. Directly connect this pin to the on-chip debug emulator or a flash memory programmer, or pull it up by connecting it to EV <sub>DD0</sub> or EV <sub>DD1</sub> via an external resistor.	p.40 <input type="checkbox"/>
				ANI8/P150 to ANI15/P157	ANI8/P150 to ANI15/P157 are set in the digital input (general-purpose port) mode after release of reset.	p.46 <input type="checkbox"/>
				REGC	Keep the wiring length as short as possible for the broken-line part in the above figure.	p.48 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Caution	Page	
Chapter 3	Soft	Memory space	PMC: Processor mode control register	Set PMC only once during the initial settings prior to operating the DMA controller. Rewriting PMC other than during the initial settings is prohibited.	p.66	<input type="checkbox"/>
				After setting PMC, wait for at least one instruction and access the mirror area.	p.66	<input type="checkbox"/>
			Internal data memory space	It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.	p.66	<input type="checkbox"/>
				While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory. Furthermore, the areas of F8700H to F8EFFH cannot be used with the $\mu$ PD78F1178 and $\mu$ PD78F1178A, respectively.	p.66	<input type="checkbox"/>
			SFR: Special function register area	Do not access addresses to which SFRs are not assigned.	pp.67, 78	<input type="checkbox"/>
		2nd SFR: Extended special function register	Do not access addresses to which the 2nd SFR is not assigned.	pp.67, 84	<input type="checkbox"/>	
		Processor registers	SP: Stack pointer	Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.	p.74	<input type="checkbox"/>
				The values of the stack pointer must be set to even numbers. If odd numbers are specified, the least significant bit is automatically cleared to 0.	p.74	<input type="checkbox"/>
				It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as a stack area.	p.74	<input type="checkbox"/>
				While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory. Furthermore, the areas of F8700H to F8EFFH cannot be used with the $\mu$ PD78F1178 and $\mu$ PD78F1178A, respectively.	p.74	<input type="checkbox"/>
		General-purpose registers	It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.	p.75	<input type="checkbox"/>	
Chapter 4	Soft	Port functions	P01/TO00	To use P01/TO00 as a general-purpose port, set bit 0 (TO00) of timer output register 0 (TO0) and bit 0 (TOE00) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.	p.109	<input type="checkbox"/>
			P02/SO10/TxD1, P03/SI10/RxD1/SDA10, P04/SCK10/SCL10	To use P02/SO10/TxD1, P03/SI10/RxD1/SDA10, or P04/SCK10/SCL10 as a general-purpose port, note the serial array unit 0 setting. For details, refer to the following tables. <ul style="list-style-type: none"> <li>• Table 13-7 Relationship Between Register Settings and Pins (Channel 2 of Unit 0: CSI10, UART1 Transmission, IIC10)</li> <li>• Table 13-8 Relationship Between Register Settings and Pins (Channel 3 of Unit 0: CSI11, UART1 Reception, IIC11)</li> </ul>	p.109	<input type="checkbox"/>
			P10/SCK00/EX24, P11/SI00/RxD0/EX25, P12/SO00/TxD0/EX26, P13/TxD3/EX27, P14/RxD3/EX28	To use P10/SCK00/EX24, P11/SI00/RxD0/EX25, P12/SO00/TxD0/EX26, P13/TxD3/EX27, or P14/RxD3/EX28 as a general-purpose port, note the serial array unit setting. For details, refer to the following tables. <ul style="list-style-type: none"> <li>• Table 13-5 Relationship Between Register Settings and Pins (Channel 0 of Unit 0: CSI00, UART0 Transmission)</li> <li>• Table 13-6 Relationship Between Register Settings and Pins (Channel 1 of Unit 0: CSI01, UART0 Reception)</li> <li>• Table 13-11 Relationship Between Register Settings and Pins (Channel 2 of Unit 1: UART3 Transmission)</li> <li>• Table 13-12 Relationship Between Register Settings and Pins (Channel 3 of Unit 1: UART3 Reception)</li> </ul>	p.117	<input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 4	Soft	Port functions	P16/TI01/TO01/INTP5/EX30, P17/TI02/TO02/EX31	To use P16/TI01/TO01/INTP5/EX30 or P17/TI02/TO02/EX31 as a general-purpose port, set bits 1 and 2 (TO01, TO02) of timer output register 0 (TO0) and bits 1 and 2 (TOE01, TOE02) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.	p.117 <input type="checkbox"/>
			P15/RTCDIV/RTCC/EX29	To use P15/RTCDIV/RTCC/EX29 as a general-purpose port, set bit 4 (RCLOE0) of real-time counter control register 0 (RTCC0) and bit 6 (RCLOE2) of real-time counter control register 2 (RTCC2) to "0", which is the same as their default status settings.	p.117 <input type="checkbox"/>
			Port 1	Do not enable outputting other alternate functions when the external expansion output (address bus) function is used.	p.117 <input type="checkbox"/>
	Hard		Port 2	See 2.2.17 AV <sub>REF0</sub> for the voltage to be applied to the AV <sub>REF0</sub> pin when using port 2 as a digital I/O.	p.123 <input type="checkbox"/>
			Soft	P31/TI03/TO03/INTP4	To use P31/TI03/TO03/INTP4 as a general-purpose port, set bit 3 (TO03) of timer output register 0 (TO0) and bit 3 (TOE03) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.
	Soft		P30/RTC1HZ/INTP3	To use P30/RTC1HZ/INTP3 as a general-purpose port, set bit 5 (RCLOE1) of real-time counter control register 0 (RTCC0) to "0", which is the same as its default status setting.	p.125 <input type="checkbox"/>
			P40/TOOL0, P41/TOOL1	When a tool is connected, the P40 pin cannot be used as a port pin. When the on-chip debug function is used, P41 pin can be used as follows by the mode setting on the debugger. 1-line mode: can be used as a port (P41). 2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).	p.128 <input type="checkbox"/>
			P43/SCK01, P44/SI01, P45/SO01	To use P43/SCK01, P44/SI01, or P45/SO01 as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 13-6 Relationship Between Register Settings and Pins (Channel 1 of Unit 0: CSI01, UART0 reception).	p.128 <input type="checkbox"/>
			P42/TI04/TO04, P46/INTP1/TI05/TO05	To use P42/TI04/TO04 or P46/INTP1/TI05/TO05 as a general-purpose port, set bits 4 and 5 (TO04, TO05) of timer output register 0 (TO0) and bits 4 and 5 (TOE04, TOE05) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.	p.128 <input type="checkbox"/>
			P60/SCL0, P61/SDA0	Stop the operation of serial interface IIC0 when using P60/SCL0 and P61/SDA0 as generalpurpose ports.	p.139 <input type="checkbox"/>
			P95/SCK11/SCL11, P96/SI11/SDA11, P97/SO11	To use P95/SCK11/SCL11, P96/SI11/SDA11, or P97/SO11 as a general-purpose port, note the serial array unit setting. For details, refer to Table 13-8 Relationship Between Register Settings and Pins (Channel 3 of Unit 0: CSI11, UART1 Reception, IIC11).	p.144 <input type="checkbox"/>
			P110, P111	See 2.2.18 AV <sub>REF1</sub> for the voltage to be applied to the AV <sub>REF1</sub> pin when using P110 and P111 as digital I/O.	p.148 <input type="checkbox"/>
			P121 to P124	The function setting on P121 to P124 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.	p.150 <input type="checkbox"/>
			P125/SCK21/SCL21, P126/SI21/SDA21, or P127/SO21	To use P125/SCK21/SCL21, P126/SI21/SDA21, or P127/SO21 as a general-purpose port, note the serial array unit 1 setting. For details, refer to Table 13-10 Relationship Between Register Settings and Pins (Channel 1 of Unit 1: CSI21, UART2 reception, IIC21).	p.150 <input type="checkbox"/>
			P131/TI06/TO06	To use P131/TI06/TO06 as a general-purpose port, set bit 6 (TO06) of timer output register 0 (TO0) and bit 6 (TOE06) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.	p.155 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page		
Chapter 4	Soft	Port functions	P142/SCK20/SCL20, P143/SI20/RxD2/SDA20, P144/SO20/TxD2	To use P142/SCK20/SCL20, P143/SI20/RxD2/SDA20, or P144/SO20/TxD2 as a general-purpose port, note the serial array unit 1 setting. For details, refer to the following tables. <ul style="list-style-type: none"> <li>• Table 13-9 Relationship Between Register Settings and Pins (Channel 0 of Unit 1: CSI20, UART2 Transmission, IIC20)</li> <li>• Table 13-10 Relationship Between Register Settings and Pins (Channel 1 of Unit 1: CSI21, UART2 Reception, IIC21)</li> </ul>	p.157	<input type="checkbox"/>	
			P145/TI07/TO07	To use P145/TI07/TO07 as a general-purpose port, set bit 7 (TO07) of timer output register 0 (TO0) and bit 7 (TOE07) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.	p.157	<input type="checkbox"/>	
			P140/PCLBUZ0/INTP6, P141/PCLBUZ1/INTP7	To use P140/PCLBUZ0/INTP6 or P141/PCLBUZ1/INTP7 as a general-purpose port, set bit 7 of clock output select registers 0 and 1 (CKS0, CKS1) to "0", which is the same as their default status settings.	p.157	<input type="checkbox"/>	
		Hard	Port 15		See 2.2.17 AV <sub>REF0</sub> for the voltage to be applied to the AV <sub>REF0</sub> pin when using port 15 as a digital I/O.	p.161	<input type="checkbox"/>
		Soft	P160/TI10/TO10 to P163/TI13/TO13	To use P160/TI10/TO10 to P163/TI13/TO13 as a general-purpose port, set bits 0 to 3 (TO10 to TO13) of timer output register 0 (TO0) and bits 0 to 3 (TOE10 to TOE13) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.	p.162	<input type="checkbox"/>	
	PM0 to PM9, PM11 to PM16: Port mode registers		Be sure to set bits 1 to 4 of PM12, bits 2 to 7 of PM13, bits 6 and 7 of PM14, and bits 4 to 7 of PM16 to "1". And be sure to set bit 0 of PM13 to "0".	p.164	<input type="checkbox"/>		
	ADPC: A/D port configuration register		Set the channel used for A/D conversion to the input mode by using port mode registers 2 and 15 (PM2, PM15).	p.171	<input type="checkbox"/>		
			Do not set the pin set by ADPC as digital I/O by analog input channel specification register (ADS).	p.171	<input type="checkbox"/>		
			P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 are set as analog inputs in the order of P157/ANI15, ..., P150/ANI8, P27/ANI7, ..., P20/ANI0 by the A/D port configuration register (ADPC). When using P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 as analog inputs, start designing from P157/ANI15.	p.171	<input type="checkbox"/>		
	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.	p.179	<input type="checkbox"/>		
	Chapter 5		Soft	External bus interface	PER1: Peripheral enable register 1	When setting the external bus interface, be sure to set EXBEN to 1 first. If EXBEN = 0, writing to a control register of the external bus interface is ignored, and, even if the register is read, only the default value is read (except for port mode registers 0, 1, 5, 6, 7, 8, 9 (PM0, PM1, PM5, PM6, PM7, PM8, PM9) and port registers 0, 1, 5, 6, 7, 8, 9 (P0, P1, P5, P6, P7, P8, P9)).	p.185
		Number of instruction execution clocks and instruction wait clocks for fetch access			The flash memory and external memory are located in consecutive spaces, but start fetching in the external memory space by using a branch instruction (CALL, BR) in the flash memory or RAM memory.	p.189	<input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Caution	Page		
Chapter 6	Soft	Clock generator	CMC: Clock operation mode control register	CMC can be written only once after reset release, by an 8-bit memory manipulation instruction.	p.206 <input type="checkbox"/>		
				After reset release, set CMC before X1 or XT1 oscillation is started as set by the clock operation status control register (CSC).	p.206 <input type="checkbox"/>		
				Be sure to set AMPH to 1 if the X1 clock oscillation frequency exceeds 10 MHz.	p.206 <input type="checkbox"/>		
				It is recommended to set the default value (00H) to CMC after reset release, even when the register is used at the default value, in order to prevent malfunctioning during a program loop.	p.206 <input type="checkbox"/>		
			CSC: Clock operation status control register	After reset release, set the clock operation mode control register (CMC) before starting X1 oscillation as set by MSTOP or XT1 oscillation as set by XTSTOP.	p.207 <input type="checkbox"/>		
				To start X1 oscillation as set by MSTOP, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).	p.207 <input type="checkbox"/>		
				Do not stop the clock selected for the CPU/peripheral hardware clock (f <sub>CLK</sub> ) with the CSC register.	p.207 <input type="checkbox"/>		
				The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) and the condition before clock oscillation is to be stopped are as shown in Figure 6-2.	p.207 <input type="checkbox"/>		
			OSTC: Oscillation stabilization time counter status register	After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.	p.209 <input type="checkbox"/>		
				The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTC. In the following cases, set the oscillation stabilization time of OSTC to the value greater than or equal to the count value which is to be checked by the OSTC register. <ul style="list-style-type: none"> <li>If the X1 clock starts oscillation while the internal high-speed oscillation clock or subsystem clock is being used as the CPU clock.</li> <li>If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock with the X1 clock oscillating. (Note, therefore, that only the status up to the oscillation stabilization time set by OSTC is set to OSTC after the STOP mode is released.)</li> </ul>	p.209 <input type="checkbox"/>		
			Hard	Soft	OSTS: Oscillation stabilization time select register	The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).	p.209 <input type="checkbox"/>
						To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTC register before executing the STOP instruction.	p.211 <input type="checkbox"/>
	Setting the oscillation stabilization time to 20 μs or less is prohibited.	p.211 <input type="checkbox"/>					
	To change the setting of the OSTC register, be sure to confirm that the counting operation of the OSTC register has been completed.	p.211 <input type="checkbox"/>					
	Do not change the value of the OSTC register during the X1 clock oscillation stabilization time.	p.211 <input type="checkbox"/>					
	The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTC. In the following cases, set the oscillation stabilization time of OSTC to the value greater than or equal to the count value which is to be checked by the OSTC register. <ul style="list-style-type: none"> <li>If the X1 clock starts oscillation while the internal high-speed oscillation clock or subsystem clock is being used as the CPU clock.</li> <li>If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock with the X1 clock oscillating. (Note, therefore, that only the status up to the oscillation stabilization time set by OSTC is set to OSTC after the STOP mode is released.)</li> </ul>	p.211 <input type="checkbox"/>					

Chapter	Classification	Function	Details of Function	Cautions	Page		
Chapter 6	Hard	Clock generator	OSTS: Oscillation stabilization time select register	The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).	p.211 <input type="checkbox"/>		
				Soft	CKC: System clock control register	Be sure to set bit 3 to 1.	p.213 <input type="checkbox"/>
						The clock set by CSS, MCM0, and MDIV2 to MDIV0 is supplied to the CPU and peripheral hardware. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the real-time counter, clock output/buzzer output, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral operating hardware clock.	p.213 <input type="checkbox"/>
	Hard			If the peripheral hardware clock is used as the subsystem clock, the operations of the A/D converter and IIC0 are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS).	p.213 <input type="checkbox"/>		
				Soft	PER0, PER1: Peripheral enable registers 0, 1	OSMC: Operation speed mode control register	Be sure to clear bits 1 to 7 of the PER1 register to 0.
	OSMC can be written only once after reset release, by an 8-bit memory manipulation instruction.	p.216 <input type="checkbox"/>					
	Write "1" to FSEL before the following two operations. • Changing the clock prior to dividing $f_{CLK}$ to a clock other than $f_{IH}$ . • Operating the DMA controller.	p.216 <input type="checkbox"/>					
	The CPU waits when "1" is written to the FSEL flag. Interrupt requests issued during a wait will be suspended. The wait time is 16.6 $\mu$ S to 18.5 $\mu$ S when $f_{CLK} = f_{IH}$ , and 33.3 $\mu$ S to 36.9 $\mu$ S when $f_{CLK} = f_{IH}/2$ . However, counting the oscillation stabilization time of $f_x$ can continue even while the CPU is waiting.	p.216 <input type="checkbox"/>					
	To increase $f_{CLK}$ to 10 MHz or higher, set FSEL to "1", then change $f_{CLK}$ after two or more clocks have elapsed. Use the external bus interface two clock cycles after setting FSEL to 1.	p.216 <input type="checkbox"/>					
	Soft	HIOTRM: Internal high-speed oscillator trimming register		Flash memory can be used at a frequency of 10 MHz or lower if FSEL is 1.	p.216 <input type="checkbox"/>		
				The frequency will vary if the temperature and $V_{DD}$ pin voltage change after accuracy adjustment. Moreover, if the HIOTRM register is set to any value other than the initial value (10H), the oscillation accuracy of the internal high-speed oscillation clock may exceed 8 MHz $\pm$ 5%, depending on the subsequent temperature and $V_{DD}$ voltage change, or HIOTRM register setting. When the temperature and $V_{DD}$ voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.	p.217 <input type="checkbox"/>		
				The internal high-speed oscillation frequency becomes faster/slower by increasing/decreasing the HIOTRM value to a value larger/smaller than a certain value. A reversal, such as the frequency becoming slower/faster by increasing/decreasing the HIOTRM value does not occur.	p.218 <input type="checkbox"/>		



Chapter	Classification	Function	Details of Function	Cautions	Page		
Chapter 6	Hard	X1/XT1 oscillator	-	When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 6-10 and 6-11 to avoid an adverse effect from wiring capacitance.	p.220	<input type="checkbox"/>	
				<ul style="list-style-type: none"> <li>• Keep the wiring length as short as possible.</li> <li>• Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.</li> <li>• Always make the ground point of the oscillator capacitor the same potential as V<sub>ss</sub>. Do not ground the capacitor to a ground pattern through which a high current flows.</li> <li>• Do not fetch signals from the oscillator.</li> </ul> Note that the XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption.			
				When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.	p.221	<input type="checkbox"/>	
		Clock generator operation when power supply voltage is turned on	When LVI default start function stopped is set (option byte: LVIOFF = 1)	If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 1.8 V, input a low level to the RESET pin from power application until the voltage reaches 1.8 V, or set the LVI default start function stopped by using the option byte (LVIOFF = 0) (see Figure 6-14). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 6-13 after reset release by the RESET pin.	p.225	<input type="checkbox"/>	
	It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.			p.225	<input type="checkbox"/>		
		Clock generator operation when power supply voltage is turned on	When LVI default start function enabled is set (option byte: LVIOFF = 0)	A voltage oscillation stabilization time is required after the supply voltage reaches 1.59 V (TYP.). If the supply voltage rises from 1.59 V (TYP.) to 2.07 V (TYP.) within the power supply oscillation stabilization time, the power supply oscillation stabilization time is automatically generated before reset processing.	p.226	<input type="checkbox"/>	
	It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.			p.226	<input type="checkbox"/>		
		Soft	Controlling high-speed system clock	X1/P121, X2/EXCLK/P122	The X1/P121 and X2/EXCLK/P122 pins are in the input port mode after a reset release.	p.227	<input type="checkbox"/>
				X1 clock	The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. Therefore, it is necessary to also set the value of the OSCSELS bit at the same time. For OSCSELS bit, see 6.6.3 Example of controlling subsystem clock.	p.227	<input type="checkbox"/>
					Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)).	p.227	<input type="checkbox"/>
				External main system clock	The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. Therefore, it is necessary to also set the value of the OSCSELS bits at the same time. For OSCSELS bits, see 6.6.3 Example of controlling subsystem clock.	p.228	<input type="checkbox"/>
				Set the external main system clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)).	p.228	<input type="checkbox"/>	

Chapter	Classification	Function	Details of Function	Cautions	Page	
Chapter 6	Soft	Controlling high-speed system clock	High-speed system clock	Be sure to clear bits 1 to 7 of PER1 to 0.	p.229 <input type="checkbox"/>	
				Be sure to confirm that MCS = 0 or CLS = 1 when setting MSTOP to 1. In addition, stop peripheral hardware that is operating on the high-speed system clock.	p.230 <input type="checkbox"/>	
		Controlling internal high-speed oscillation clock	Internal high-speed oscillation clock	If switching the CPU/peripheral hardware clock from the high-speed system clock to the internal high-speed oscillation clock after restarting the internal high-speed oscillation clock, do so after 10 $\mu$ s or more have elapsed. If the switching is made immediately after the internal high-speed oscillation clock is restarted, the accuracy of the internal high-speed oscillation cannot be guaranteed for 10 $\mu$ s.	p.231 <input type="checkbox"/>	
				Be sure to confirm that MCS = 1 or CLS = 1 when setting HIOSTOP to 1. In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock.	p.232 <input type="checkbox"/>	
	Hard	Subsystem clock control	XT1/P123, XT2/P124 Subsystem clock	The XT1/P123 and XT2/P124 pins are in the input port mode after a reset release.	p.232 <input type="checkbox"/>	
				When the subsystem clock is used as the CPU clock, the subsystem clock is also supplied to the peripheral hardware (except the real-time counter, clock output/buzzer output, and watchdog timer). At this time, the operations of the A/D converter and IIC0 are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS).	pp.232, 233 <input type="checkbox"/>	
		Soft	Subsystem clock control	Subsystem clock	The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. Therefore, it is necessary to also set the value of the EXCLK and OSCSEL bits at the same time. For EXCLK and OSCSEL bits, see 6.6.1 (1) Example of setting procedure when oscillating the X1 clock or 6.6.1 (2) Example of setting procedure when using the external main system clock.	p.232 <input type="checkbox"/>
					Be sure to confirm that CLS = 0 when setting XTSTOP to 1. In addition, stop the peripheral hardware if it is operating on the subsystem clock.	p.233 <input type="checkbox"/>
					The subsystem clock oscillation cannot be stopped using the STOP instruction.	p.233 <input type="checkbox"/>
					CPU clock status transition	–
Chapter 7	Soft	Timer array unit	TCRmn: Timer/counter register mn	The count value is not captured to TDRmn even when TCRmn is read.	p.249 <input type="checkbox"/>	
			TDRmn: Timer data register mn	TDRmn does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.	p.251 <input type="checkbox"/>	
			PER0: Peripheral enable register 0	When setting the timer array unit, be sure to set TAUmEN to 1 first. If TAUmEN = 0, writing to a control register of the timer array unit is ignored, and all read values are default values (except for timer input select register m (TISm), input switch control register (ISC), noise filter enable registers 1, 2 (NFEN1, 2), port mode registers 0, 1, 3, 4, 13, 14, 16 (PM0, PM1, PM3, PM4, PM13, PM14, PM16), and port registers 0, 1, 3, 4, 13, 14, 16 (P0, P1, P3, P4, P13, P14, P16)).	p.253 <input type="checkbox"/>	

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 7	Soft	Timer array unit	TPSm: Timer clock select register m	Be sure to clear bits 15 to 8 to "0".	p.255 <input type="checkbox"/>
			TMRmn: Timer mode register mn	Be sure to clear bits 14, 13, 5, and 4 to "0".	p.256 <input type="checkbox"/>
			TSm: Timer channel start register m	Be sure to clear bits 15 to 8 of TS0 and bits 15 to 4 of TS1 to "0"	p.261 <input type="checkbox"/>
				In the first cycle operation of count clock after writing TSmn, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.	pp.262, 263 <input type="checkbox"/>
				An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TImn is used).	pp.264, 265 <input type="checkbox"/>
			TTm: Timer channel stop register m	Be sure to clear bits 15 to 8 of TT0 and bits 15 to 4 of TT1 to "0".	p.266 <input type="checkbox"/>
			TOEm: Timer output enable register m	Be sure to clear bits 15 to 8 of TOE0 and bits 15 to 4 of TOE1 to "0".	p.268 <input type="checkbox"/>
			TOm: Timer output register m	Be sure to clear bits 15 to 8 of TO0 and bits 15 to 4 of TO1 to "0".	p.269 <input type="checkbox"/>
			TOLm: Timer output level register m	Be sure to clear bits 15 to 8 of TOL0 and bits 15 to 4 of TOL1 to "0".	p.270 <input type="checkbox"/>
			TOMm: Timer output mode register m	Be sure to clear bits 15 to 8 of TOM0 and bits 15 to 4 of TOM1 to "0".	p.271 <input type="checkbox"/>
			ISC: Input switch control register	Be sure to clear bits 7 to 2 to "0".	p.272 <input type="checkbox"/>
			Channel output (TOmn pin) operation	(1) Changing values set in registers TOm, TOEm, TOLm, and TOMm during timer operation Since the timer operations (operations of TCRmn and TDRmn) are independent of the TOmn output circuit and changing the values set in TOm, TOEm, TOLm, and TOMm does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TOmn pin by timer operation, however, set TOm, TOEm, TOLm, and TOMm to the values stated in the register setting example of each operation. When the values set in TOEm, TOLm, and TOMm (except for TOm) are changed close to the timer interrupt (INTTMmn), the waveform output to the TOmn pin may be different depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMmn) signal generation timing.	p.278 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page	
Chapter 7	Soft	Timer array unit	Channel output (TOmn pin) operation	(2) Default level of TOmn pin and output level after timer operation start The following figure shows the TOmn pin output level transition when writing has been done in the state of TOEmn = 0 before port output is enabled and TOEmn = 1 is set after changing the default level. (a) When operation starts with TOMmn = 0 setting (toggle output) The setting of TOLmn is invalid when TOMmn = 0. When the timer operation starts after setting the default level, the toggle signal is generated and the output level of TOmn pin is reversed. (b) When operation starts with TOMmn = 1 setting (combination operation mode (PWM output)) When TOMmn = 1, the active level is determined by TOLmn setting.	pp.278, 279 <input type="checkbox"/>	
				(3) Operation of TOmn pin in combination operation mode (TOMmn = 1) (a) When TOLmn setting has been changed during timer operation When the TOLmn setting has been changed during timer operation, the setting becomes valid at the generation timing of TOmn change condition. Rewriting TOLmn does not change the output level of TOmn. The following figure shows the operation when the value of TOLmn has been changed during timer operation (TOMmn = 1). (b) Set/reset timing To realize 0%/100% output at PWM output, the TOmn pin/TOmn set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel. If the set condition and reset condition are generated at the same time, a higher priority is given to the latter. Figure 7-31 shows the set/reset operating statuses where the master/slave channels are set as follows.	pp.279, 280 <input type="checkbox"/>	
				Collective manipulation of TOmn bits	When TOEmn = 1, even if the output by timer interrupt of each timer (INTTMmn) contends with writing to TOmn, output is normally done to TOmn pin.	p.282 <input type="checkbox"/>
		Operation of timer array unit as independent channel	Input pulse interval measurement	Input signal high-/low-level width measurement	The TImn pin input is sampled using the operating clock selected with the CKSmn bit of the TMRmn register, so an error equal to the number of operating clocks occurs.	p.301 <input type="checkbox"/>
				PWM function	To rewrite both TDRmn of the master channel and TDRmp of the slave channel, a write access is necessary two times. The timing at which the values of TDRmn and TDRmp are loaded to TCRmn and TRCmp is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both TDRmn of the master and TDRmp of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.	p.309 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 7	Soft	Operation of plural channels of timer array unit	One-shot pulse output function	The timing of loading of TDRmn of the master channel is different from that of TDRmp of the slave channel. If TDRmn and TDRmp are rewritten during operation, therefore, an illegal waveform is output. Rewrite the TDRmn after INTTMmn is generated and the TDRmp after INTTMmp is generated.	p.316 <input type="checkbox"/>
			Multiple PWM output function	To rewrite both TDRmn of the master channel and TDRmp of the slave channel 1, write access is necessary at least twice. Since the values of TDRmn and TDRmp are loaded to TCRmn and TCRmp after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both TDRmn of the master and TDRmp of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to TDRmq of the slave channel 2).	p.323 <input type="checkbox"/>
Chapter 8	Soft	Real-time counter	PER0: Peripheral enable register 0	When using the real-time counter, first set RTCEN to 1, while oscillation of the subsystem clock ( $f_{SUB}$ ) is stable. If RTCEN = 0, writing to a control register of the real-time counter is ignored, and, even if the register is read, only the default value is read.	p.334 <input type="checkbox"/>
			RTCC0: Real-time counter control register 0	If RCLOE0 and RCLOE1 are changed when RTCE = 1, glitches may occur in the 32.768 kHz and 1 Hz output signals.	p.335 <input type="checkbox"/>
			RTCC1: Real-time counter control register 1	The RIFG and WAFG flags may be cleared when the RTCC1 register is written by using a 1-bit manipulation instruction. Use, therefore, an 8-bit manipulation instruction in order to write to the RTCC1 register. To prevent the RIFG and WAFG flags from being cleared during writing, disable writing by setting "1" to the corresponding bit. When the value may be rewritten because the RIFG and WAFG flags are not being used, the RTCC1 register may be written by using a 1-bit manipulation instruction.	p.337 <input type="checkbox"/>
			RTCC2: Real-time counter control register 2	Change ICT2, ICT1, and ICT0 when RINTE = 0.	p.338 <input type="checkbox"/>
				When the output from RTCDIV pin is stopped, the output continues after a maximum of two clocks of $f_{XT}$ and enters the low level. While 512 Hz is output, and when the output is stopped immediately after entering the high level, a pulse of at least one clock width of $f_{XT}$ may be generated.	p.338 <input type="checkbox"/>
			RSUBC: Sub-count register	After the real-time counter starts operating, the output width of the RTCDIV pin may be shorter than as set during the first interval period.	p.338 <input type="checkbox"/>
				When a correction is made by using the SUBCUJ register, the value may become 8000H or more.	p.339 <input type="checkbox"/>
				This register is also cleared by reset effected by writing the second count register.	p.339 <input type="checkbox"/>
			HOUR: Hour count register	The value read from this register is not guaranteed if it is read during operation, because a value that is changing is read.	p.339 <input type="checkbox"/>
				Bit 5 (HOUR20) of HOUR indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).	p.340 <input type="checkbox"/>
WEEK: Week count register	The value corresponding to the month count register or the day count register is not stored in the week count register automatically. After reset release, set the week count register as follow.	p.343 <input type="checkbox"/>			
ALARMWM: Alarm minute register	Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.	p.346 <input type="checkbox"/>			

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 8	Soft	Real-time counter	ALARMWH: Alarm hour register	Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.	p.346 <input type="checkbox"/>
				Bit 5 (WH20) of ALARMWH indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).	p.346 <input type="checkbox"/>
			Reading/writing real-time counter	Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.	pp.350, 351 <input type="checkbox"/>
			1, 512 Hz and 32.768, 16.384 kHz outputs of real-time counter	First set RTCEN to 1, while oscillation of the subsystem clock ( $f_{SUB}$ ) is stable.	p.353 <input type="checkbox"/>
Chapter 9	Soft	Watchdog timer	WDTE: Watchdog timer enable register	If a value other than "ACH" is written to WDTE, an internal reset signal is generated.	p.361 <input type="checkbox"/>
				If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated.	p.361 <input type="checkbox"/>
				The value read from WDTE is 9AH/1AH (this differs from the written value (ACH)).	p.361 <input type="checkbox"/>
		Controlling operation	When data is written to WDTE for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.	p.362 <input type="checkbox"/>	
			If the watchdog timer is cleared by writing "ACH" to WDTE, the actual overflow time may be different from the overflow time set by the option byte by up to 2/ $f_{L}$ seconds.	p.362 <input type="checkbox"/>	
			The watchdog timer can be cleared immediately before the count value overflows.	p.362 <input type="checkbox"/>	
			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). (See the table on page 361.) If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts. When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed. Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset. Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.	p.363 <input type="checkbox"/>	
			The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.	p.363 <input type="checkbox"/>	
			Setting overflow time	The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.	p.363 <input type="checkbox"/>
			Setting window open period	When data is written to WDTE for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.	p.364 <input type="checkbox"/>
	The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.	p.364 <input type="checkbox"/>			

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 9	Soft	Watchdog timer	Setting window open period	When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of WINDOW1 and WINDOW0.	p.364 <input type="checkbox"/>
				Do not set the window open period to 25% if the watchdog timer corresponds to either of the conditions below. <ul style="list-style-type: none"> <li>• When used at a supply voltage (<math>V_{DD}</math>) below 2.7 V.</li> <li>• When stopping all main system clocks (internal high-speed oscillation clock, X1 clock, and external main system clock) by use of the STOP mode or software.</li> <li>• Low-power consumption mode</li> </ul>	p.364 <input type="checkbox"/>
			Setting interval interrupt	When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed. Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset. Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.	p.365 <input type="checkbox"/>
Chapter 10	Soft	Clock output/buzzer output controller	CKSn: Clock output select registers n	Change the output clock after disabling clock output (PCLOEn = 0). If the selected clock ( $f_{MAIN}$ or $f_{SUB}$ ) stops during clock output (PCLOEn = 1), the output becomes undefined.	p.368 <input type="checkbox"/> p.368 <input type="checkbox"/>
Chapter 11	Soft	A/D converter	PER0: Peripheral enable register 0	When setting the A/D converter, be sure to set ADCEN 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 port mode registers 2 and 15 (PM2, PM15)).	p.373 <input type="checkbox"/>
			ADM: A/D converter mode register	A/D conversion must be stopped before rewriting bits FR0 to FR2, LV1, and LV0 to values other than the identical data.	p.374 <input type="checkbox"/>
			A/D conversion time selection ( $2.7\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ )	Set the conversion times with the following conditions. Conventional-specification products ( $\mu\text{PD78F117x}$ ) <ul style="list-style-type: none"> <li>• <math>4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}</math>: <math>f_{AD} = 0.6</math> to <math>3.6\text{ MHz}</math></li> <li>• <math>2.7\text{ V} \leq AV_{REF0} &lt; 4.0\text{ V}</math>: <math>f_{AD} = 0.6</math> to <math>1.8\text{ MHz}</math></li> </ul> Functionally expanded products ( $\mu\text{PD78F117xA}$ ) <ul style="list-style-type: none"> <li>• <math>4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}</math>: <math>f_{AD} = 0.33</math> to <math>3.6\text{ MHz}</math></li> <li>• <math>2.7\text{ V} \leq AV_{REF0} &lt; 4.0\text{ V}</math>: <math>f_{AD} = 0.33</math> to <math>1.8\text{ MHz}</math></li> </ul>	p.375 <input type="checkbox"/>
			A/D conversion time selection ( $2.3\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}$ )	Set the conversion times with the following conditions. <ul style="list-style-type: none"> <li>• <math>4.0\text{ V} \leq AV_{REF0} \leq 5.5\text{ V}</math>: <math>f_{AD} = 0.6</math> to <math>3.6\text{ MHz}</math></li> <li>• <math>2.7\text{ V} \leq AV_{REF0} &lt; 4.0\text{ V}</math>: <math>f_{AD} = 0.6</math> to <math>1.8\text{ MHz}</math></li> <li>• <math>2.3\text{ V} \leq AV_{REF0} &lt; 2.7\text{ V}</math>: <math>f_{AD} = 0.6</math> to <math>1.44\text{ MHz}</math></li> </ul>	p.376 <input type="checkbox"/>
				When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.	p.376 <input type="checkbox"/>
				Change LV1 and LV0 from the default value, when $2.3\text{ V} \leq AV_{REF0} < 2.7\text{ V}$ .	p.376 <input type="checkbox"/>
				The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.	p.376 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 11	Soft	A/D converter	ADCR: 10-bit A/D conversion result register	When writing to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.	p.378 <input type="checkbox"/>
			ADCRH: 8-bit A/D conversion result register	When writing to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.	p.378 <input type="checkbox"/>
			ADS: Analog input channel specification register	Be sure to clear bits 4 to 6 to "0".	p.379 <input type="checkbox"/>
				Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 15 (PM2, PM15).	p.379 <input type="checkbox"/>
				Do not set the pin that is set by ADPC as digital I/O by ADS.	p.379 <input type="checkbox"/>
			ADPC: A/D port configuration register	Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 15 (PM2, PM15).	p.380 <input type="checkbox"/>
				Do not set the pin that is set by ADPC as digital I/O by ADS.	p.380 <input type="checkbox"/>
				P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 are set as analog inputs in the order of P157/ANI15, ..., P150/ANI8, P27/ANI7, ..., P20/ANI0 by the A/D port configuration register (ADPC). When using P20/ANI0 to P27/ANI7 and P150/ANI8 to P157/ANI15 as analog inputs, start designing from P157/ANI15.	p.380 <input type="checkbox"/>
			PM2, PM15: Port mode registers 2 and 15	If a pin is set as an analog input port, not the pin level but "0" is always read.	p.381 <input type="checkbox"/>
			Basic operations of A/D converter	Make sure the period of <2> to <6> is 1 $\mu$ s or more.	p.382 <input type="checkbox"/>
			A/D conversion operation	Make sure the period of <2> to <6> is 1 $\mu$ s or more.	p.386 <input type="checkbox"/>
				<2> may be done between <3> and <5>.	p.386 <input type="checkbox"/>
				The period from <7> to <10> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM. The period from <9> to <10> is the conversion time set using FR2 to FR0, LV1, and LV0.	p.386 <input type="checkbox"/>
			Temperature sensor function	The temperature sensor cannot be used when low current consumption mode is set (RMC = 5AH) or when the internal high-speed oscillator has been stopped (HIOSTOP = 1 (bit 0 of CSC register)). The temperature sensor can operate as long as the internal high-speed oscillator operates (HIOSTOP = 0), even if it is not selected as the CPU/peripheral hardware clock source.	p.387 <input type="checkbox"/>
			Registers used by temperature sensors	Setting of the A/D port configuration register (ADPC), port mode register 2 (PM2) and port register 2 (P2) is not required when using the temperature sensor. There is no problem if the pin function is set as digital I/O.	p.388 <input type="checkbox"/>
				Set the conversion times so as to satisfy the following condition. $f_{AD} = 0.6$ to 1.8 MHz	p.388 <input type="checkbox"/>
When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion (ADCS = 0) beforehand.	p.388 <input type="checkbox"/>				
The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.	p.388 <input type="checkbox"/>				



Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 11	Soft	A/D converter	Registers used by temperature sensors	When using a temperature sensor, use the result of the second or later A/D conversion for temperature sensor 0 (ANI0 side), and the result of the third or later A/D conversion for temperature sensor 1 (ANI1 side).	p.389 <input type="checkbox"/>
				Be sure to clear bits 4 to 6 to "0".	p.389 <input type="checkbox"/>
		Procedure for using temperature sensors	Make sure the period of <2> to <5> is 1 $\mu$ s or more. If ADCS is set to 1 within 1 $\mu$ s, the result of the third and later conversion becomes valid on the sensor 0 side.	p.393 <input type="checkbox"/>	
			<2> can be done between <3> and <4>.	p.393 <input type="checkbox"/>	
			The period from <7> to <10> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM. The period from <9> to <10> is the conversion time set using FR2 to FR0, LV1, and LV0.	p.393 <input type="checkbox"/>	
			Do not change the AV <sub>REF0</sub> voltage during <4> to <13>. Although the temperature sensor detection value does not depend on the AV <sub>REF0</sub> voltage and thus there is no problem even if the AV <sub>REF0</sub> voltage varies at every temperature measurement, it must be stable during a measurement cycle (from <4> to <13>).	p.393 <input type="checkbox"/>	
		Use the result of the second or later A/D conversion for temperature sensor 0 (ANI0 side), and the result of the third or later A/D conversion for temperature sensor 1 (ANI1 side).	p.394 <input type="checkbox"/>		
		Operating current in STOP mode	Shift to STOP mode after clearing the A/D converter (by clearing bit 7 (ADCS) of the A/D converter mode register (ADM) to 0). The operating current can be reduced by clearing bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 at the same time. To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1L (IF1L) to 0 and start operation.	p.397 <input type="checkbox"/>	
		Reducing current when A/D converter is stopped	Be sure that the voltage to be applied to AV <sub>REF0</sub> normally satisfies the conditions stated in Table 11-1. If bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) are set to 0, the current will not be increased by the A/D converter even if a voltage is applied to AV <sub>REF0</sub> , while the A/D converter is stopped. If a current flows from the power supply that supplies a voltage to AV <sub>REF0</sub> to an external circuit of the microcontroller as shown in Figure 11-25, AV <sub>REF0</sub> = 0 V = AV <sub>SS</sub> can be achieved and the external current can be reduced by satisfying the following conditions (see the main text).	p.397 <input type="checkbox"/>	
		Hard	Input range of ANI0 to ANI15	Observe the rated range of the ANI0 to ANI15 input voltage. If a voltage of AV <sub>REF0</sub> or higher and AV <sub>SS</sub> or lower (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.	p.398 <input type="checkbox"/>
	Soft	Conflicting operations	Conflict between A/D conversion result register (ADCR, ADCRH) write and ADCR or ADCRH read by instruction upon the end of conversion ADCR or ADCRH read has priority. After the read operation, the new conversion result is written to ADCR or ADCRH.	p.398 <input type="checkbox"/>	
	Conflict between ADCR or ADCRH write and A/D converter mode register (ADM) write, analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion ADM, ADS, or ADPC write has priority. ADCR or ADCRH write is not performed, nor is the conversion end interrupt signal (INTAD) generated.		p.398 <input type="checkbox"/>		

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 11	Hard	A/D converter	Noise countermeasures	To maintain the 10-bit resolution, attention must be paid to noise input to the AV <sub>REF0</sub> pin and pins ANI0 to ANI15. <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 11-26 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.	p.398 <input type="checkbox"/>
			ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157	The analog input pins (ANI0 to ANI7) are also used as input port pins (P20 to P27). The analog input pins (ANI8 to ANI15) are also used as input port pins (P150 to P157). When A/D conversion is performed with any of ANI0 to ANI15 selected, do not access P20 to P27 and P150 to P157 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to select pins used as P20 to P27 and P150 to P157 starting with the ANI0/P20 that is the furthest from AV <sub>REF0</sub> .	p.399 <input type="checkbox"/>
	Hard			If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.	p.399 <input type="checkbox"/>
			Input impedance of ANI0 to ANI15 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 10 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI15 pins (see Figure 11-26).	p.399 <input type="checkbox"/>
			AV <sub>REF0</sub> pin input impedance	A series resistor string of several tens of kΩ is connected between the AV <sub>REF0</sub> and AV <sub>SS</sub> pins. Therefore, if the output impedance of the reference voltage source is high, this will result in a series connection to the series resistor string between the AV <sub>REF0</sub> and AV <sub>SS</sub> pins, resulting in a large reference voltage error.	p.399 <input type="checkbox"/>
	Soft		Interrupt request flag (ADIF)	The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed. Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF is set despite the fact A/D conversion for the post-change analog input has not ended. When A/D conversion is stopped and then resumed, clear ADIF before the A/D conversion operation is resumed.	p.400 <input type="checkbox"/>
			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 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.	p.400 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 11	Soft	A/D converter	A/D conversion result register (ADCR, ADCRH) read operation	When a write operation is performed to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR and ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using a timing other than the above may cause an incorrect conversion result to be read.	p.400 <input type="checkbox"/>
			Internal equivalent circuit	The equivalent circuit of the analog input block is shown below (see Figure 11-28.).	p.401 <input type="checkbox"/>
			Starting the A/D converter	Start the A/D converter after the AV <sub>REF0</sub> and AV <sub>REF1</sub> voltages (the reference voltages for the D/A converter) stabilize.	p.401 <input type="checkbox"/>
Chapter 12	Soft	D/A converter	PER0: Peripheral enable register 0	When setting the D/A converter, be sure to set DACEN to 1 first. If DACEN = 0, writing to a control register of the D/A converter is ignored, and, even if the register is read, only the default value is read (except for port mode register 11 (PM11) and port register 11 (P11)).	p.404 <input type="checkbox"/>
			Operation in normal mode	Make the interval for writing DACSn of the same channel by one clock longer than f <sub>CLK</sub> . If writing is successively performed, only the value written last will be converted.	p.407 <input type="checkbox"/>
			Operation in real-time output mode	Make the interval for generating a start trigger to the same channel by one clock longer than f <sub>CLK</sub> . If a start trigger is successively generated for every f <sub>CLK</sub> , D/A conversion will be performed only at the first trigger.	p.408 <input type="checkbox"/>
				Note the following points in the procedure (i to iii) for outputting an arbitrary value in <3>. <ul style="list-style-type: none"> <li>Do not generate the start trigger of the real-time output mode before enabling D/A conversion operation in &lt;3&gt; after the value is set to the DACSn register in ii.</li> <li>An arbitrary value cannot be output in &lt;3&gt; if the DACEN bit of the PER0 register is cleared once after the value is set to the DACSn register in ii.</li> </ul>	p.408 <input type="checkbox"/>
			I/O function of digital ports alternately used as ANO0, ANO1	The digital port I/O function, which is the alternate function of the ANO0 and ANO1 pins, does not operate during D/A conversion. During D/A conversion, 0 is read from the P11 register in input mode.	p.409 <input type="checkbox"/>
			P11, PM11 registers	Do not read/write the P11 register and do not change the setting of the PM11 register during D/A conversion (otherwise the conversion accuracy may decrease).	p.409 <input type="checkbox"/>
			ANO0, ANO1 pins	It is recommended that both the ANO0 and ANO1 pins be used as analog output pins or digital I/O pins, that is, use these two channels for the same application (if these pins are used for the different applications, the conversion accuracy may decrease).	p.409 <input type="checkbox"/>
			DACSn register	In the real-time output mode, set the DACSn register value before the timer trigger is generated. In addition, do not change the set value of the DACSn register while the trigger signal is output.	p.409 <input type="checkbox"/>
			Changing operation mode	Before changing the operation mode, be sure to clear the DACEn bit of the DAM register to 0 (D/A conversion stop).	p.409 <input type="checkbox"/>
			Hard	Port alternately used as ANO0 or ANO1 pin	When using the port that functions alternately as the ANO0 or ANO1 pin, use it as the port input with few level changes.
	Applying power to and disconnecting power from AV <sub>REF1</sub> and AV <sub>REF0</sub>	Stop the conversion performed by the D/A converter when supplying AV <sub>REF1</sub> or AV <sub>REF0</sub> (the reference voltages for the A/D converter) starts or stops.		p.409 <input type="checkbox"/>	
	Soft	Reducing power consumption in STOP mode		Because the D/A converter stops operation in the STOP mode, the ANO0 and ANO1 pins go into a high impedance state, and the power consumption can be reduced. In the standby modes other than the STOP mode, however, the operation continues. To lower the power consumption, therefore, clear the DACEn bit of the DAM register to 0 (D/A conversion stop).	p.409 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 12	Hard	D/A converter	Output impedance of D/A converter	Since the output impedance of the D/A converter is high, the current cannot be obtained from the ANOn pin (n = 0, 1). When the input impedance of the load is low, insert a follower amplifier between the load and ANOn pin keeping the wiring length as short as possible (for high impedance). If the wiring becomes too long, take necessary actions such as surrounding with a ground pattern.	p.409 <input type="checkbox"/>
Chapter 13	Soft	Configuration of serial array unit	SDRmn: Lower 8 bits of the serial data register mn	Be sure to clear bit 8 to "0".	p.417 <input type="checkbox"/>
		Registers controlling serial array unit	PER0: Peripheral enable register 0	When setting serial array unit m, be sure to set SAUmEN to 1 first. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read (except for input switch control register (ISC), noise filter enable register (NFEN0), port input mode registers (PIM0, PIM4, PIM9, PIM12, PIM14), port output mode registers (POM0, POM4, POM9, POM12, POM14), port mode registers (PM0, PM1, PM4, PM9, PM12, PM14), and port registers (P0, P1, P4, P9, P12, P14)).	p.419 <input type="checkbox"/>
				After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.419 <input type="checkbox"/>
		SPSm: Serial clock select register m	Be sure to clear bits 15 to 8 to "0".	p.420 <input type="checkbox"/>	
			After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.420 <input type="checkbox"/>	
		SMRmn: Serial mode register mn	Be sure to clear bits 13 to 9, 7, 4, and 3 to "0". Be sure to set bit 5 to "1".	p.421 <input type="checkbox"/>	
		SCRmn: Serial communication operation setting register mn	Be sure to clear bits 3, 6, and 11 to "0". Be sure to set bit 2 to "1".	pp.423 to 425 <input type="checkbox"/>	
		SDRmn: Higher 7 bits of the serial data register mn	Be sure to clear bit 8 to "0".	p.426 <input type="checkbox"/>	
			Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.	p.426 <input type="checkbox"/>	
			Setting SDRmn[15:9] = 0000000B is prohibited when simplified I <sup>2</sup> C is used. Set SDRmn[15:9] to 0000001B or greater.	p.426 <input type="checkbox"/>	
			Do not write eight bits to the lower eight bits if operation is stopped (SEmn = 0). (If these bits are written to, the higher seven bits are cleared to 0.)	p.426 <input type="checkbox"/>	
		SIRmn: Serial flag clear trigger register mn	Be sure to clear bits 15 to 3 to "0".	p.429 <input type="checkbox"/>	
		SSm: Serial channel start register m	Be sure to clear bits 15 to 4 to "0".	p.431 <input type="checkbox"/>	
STm: Serial channel stop register m	Be sure to clear bits 15 to 4 to "0".	p.432 <input type="checkbox"/>			
SOEm: Serial output enable register m	Be sure to clear bits 15 to 4 of SOE0, and bits 15 to 3 of SOE1 to "0".	p.433 <input type="checkbox"/>			
SOM: Serial output register m	Be sure to set bits 11, 10, and 3 of SO1 to "1". And be sure to clear bits 15 to 12 and 7 to 4 of SOM to "0".	p.434 <input type="checkbox"/>			
SOLm: Serial output level register m	Be sure to clear bits 15 to 3 and 1 to "0".	p.435 <input type="checkbox"/>			

Chapter	Classification	Function	Details of Function	Cautions	Page		
Chapter 13	Soft	Registers controlling serial array unit	ISC: Input switch control register	Be sure to clear bits 7 to 2 to "0".	p.436 <input type="checkbox"/>		
			NFEN0: Noise filter enable register 0	Be sure to clear bits 7, 5, 3, and 1 to "0".	p.437 <input type="checkbox"/>		
		Operation stop mode	Stopping the operation by units	If SAUMEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read (except for input switch control register (ISC), noise filter enable register (NFEN0), port input mode registers (PIM0, PIM4, PIM9, PIM12, PIM14), port output mode registers (POM0, POM4, POM9, POM12, POM14), port mode registers (PM0, PM1, PM4, PM9, PM12, PM14), and port registers (P0, P1, P4, P9, P12, P14)).	p.441 <input type="checkbox"/>		
	3-wire serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) communication			Master transmission	After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	pp.447, 451, 453 <input type="checkbox"/>	
				Master transmission (in continuous transmission mode)	The MDmn0 bit 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.	p.452 <input type="checkbox"/>	
				Master reception	After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	pp.456, 459, 461 <input type="checkbox"/>	
				Master reception (in continuous reception mode)	The MDmn0 bit can be rewritten even during operation. However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.	p.460 <input type="checkbox"/>	
				Master transmission/reception	After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	pp.464, 467, 469 <input type="checkbox"/>	
				Master transmission/reception (in continuous transmission/reception mode)	The MDmn0 bit 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.	p.468 <input type="checkbox"/>	
				Slave transmission	After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	pp.472, 476, 478 <input type="checkbox"/>	
				Slave transmission (in continuous transmission mode)	The MDmn0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.	p.477 <input type="checkbox"/>	
				Slave reception	After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	pp.481, 484 <input type="checkbox"/>	
				Slave transmission/reception	Be sure to set transmit data to the SIOp register before the clock from the master is started.	pp.486, 487, 489, 491, 493 <input type="checkbox"/>	
					After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	pp.487, 491, 493 <input type="checkbox"/>	
				Slave transmission/reception (in continuous transmission/reception mode)	The MDmn0 bit 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.	p.492 <input type="checkbox"/>	
				UART (UART0, UART1, UART2, UART3) communication	-	When using serial array units 0 and 1 as UARTs, the channels of both the transmitting side (even-number channel) and the receiving side (odd-number channel) can be used only as UARTs.	p.497 <input type="checkbox"/>
					UART transmission	After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	pp.501, 505 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page		
Chapter 13	Soft	UART (UART0, UART1, UART2, UART3) communication	UART transmission (in continuous transmission mode)	The MDmn0 bit 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.	p.506 <input type="checkbox"/>		
				After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.507 <input type="checkbox"/>		
			UART reception	For the UART reception, be sure to set SMRmr of channel r that is to be paired with channel n.	pp.509, 510 <input type="checkbox"/>		
				After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	pp.511, 514 <input type="checkbox"/>		
			Calculating baud rate	Setting SDRmn [15:9] = (0000000B, 0000001B) is prohibited.	p.523 <input type="checkbox"/>		
		Simplified I <sup>2</sup> C (IIC10, IIC20) communication	Address field transmission	After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.531 <input type="checkbox"/>		
			Data reception	ACK is not output when the last data is received (NACK). Communication is then completed by setting "1" to the STmn bit to stop operation and generating a stop condition.	p.540 <input type="checkbox"/>		
			Calculating transfer rate	Setting SDRmn[15:9] = 0000000B is prohibited. Setting SDRmn[15:9] = 0000001B or more.	p.542 <input type="checkbox"/>		
		Chapter 14	Soft	Serial interface IIC0	IIC0: IIC shift register 0	Do not write data to IIC0 during data transfer.	p.558 <input type="checkbox"/>
						Write or read IIC0 only during the wait period. Accessing IIC0 in a communication state other than during the wait period is prohibited. When the device serves as the master, however, IIC0 can be written only once after the communication trigger bit (STT0) is set to 1.	p.558 <input type="checkbox"/>
PER0: Peripheral enable register 0	When setting serial interface IIC0, be sure to set IIC0EN to 1 first. If IIC0EN = 0, writing to a control register of serial interface IIC0 is ignored, and, even if the register is read, only the default value is read (except for port mode register 6 (PM6) and port register 6 (P6)).				p.561 <input type="checkbox"/>		
IICC0: IIC control register 0	The start condition is detected immediately after I <sup>2</sup> C is enabled to operate (IICE0 = 1) while the SCL0 line is at high level and the SDA0 line is at low level. Immediately after enabling I <sup>2</sup> C to operate (IICE0 = 1), set LREL0 (1) by using a 1-bit memory manipulation instruction.				p.562 <input type="checkbox"/>		
	When bit 3 (TRC0) of IIC status register 0 (IICS0) is set to 1, WREL0 is set to 1 during the ninth clock and wait is canceled, after which TRC0 is cleared and the SDA0 line is set to high impedance.				p.565 <input type="checkbox"/>		
IICF0: IIC flag register 0	Write to STCEN only when the operation is stopped (IICE0 = 0).				p.569 <input type="checkbox"/>		
	As the bus release status (IICBSY = 0) is recognized regardless of the actual bus status when STCEN = 1, when generating the first start condition (STT0 = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.				p.569 <input type="checkbox"/>		
IICRSV only when the operation is stopped (IICE0 = 0).	p.569 <input type="checkbox"/>						
IICX0: IIC function expansion register 0	Determine the transfer clock frequency of I <sup>2</sup> C by using CLX0, SMC0, CL01, and CL00 before enabling the operation (by setting bit 7 (IICE0) of IIC control register 0 (IICC0) to 1). To change the transfer clock frequency, clear IICE0 once to 0.	p.571 <input type="checkbox"/>					
Setting transfer clock	Determine the transfer clock frequency of I <sup>2</sup> C by using CLX0, SMC0, CL01, and CL00 before enabling the operation (by setting bit 7 (IICE0) of IIC control register 0 (IICC0) to 1). To change the transfer clock frequency, clear IICE0 once to 0.	p.577 <input type="checkbox"/>					

Chapter	Classification	Function	Details of Function	Cautions	Page	
Chapter 14	Soft	Serial interface IIC0	When STCEN = 0	Immediately after I <sup>2</sup> C operation is enabled (IICE0 = 1), the bus communication status (IICBSY (bit 6 of IICF0) = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication. When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected). Use the following sequence for generating a stop condition. <1> Set IIC clock select register 0 (IICCL0). <2> Set bit 7 (IICE0) of IIC control register 0 (IICC0) to 1. <3> Set bit 0 (SPT0) of IICC0 to 1.	p.591	<input type="checkbox"/>
			When STCEN = 1	Immediately after I <sup>2</sup> C operation is enabled (IICE0 = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT0 (bit 1 of IIC control register 0 (IICC0)) = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.	p.591	<input type="checkbox"/>
			If other I <sup>2</sup> C communications are already in progress	If I <sup>2</sup> C operation is enabled and the device participates in communication already in progress when the SDA0 pin is low and the SCL0 pin is high, the macro of I <sup>2</sup> C recognizes that the SDA0 pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, $\overline{\text{ACK}}$ is returned, but this interferes with other I <sup>2</sup> C communications. To avoid this, start I <sup>2</sup> C in the following sequence. <1> Clear bit 4 (SPIE0) of IICC0 to 0 to disable generation of an interrupt request signal (INTIIC0) when the stop condition is detected. <2> Set bit 7 (IICE0) of IICC0 to 1 to enable the operation of I <sup>2</sup> C. <3> Wait for detection of the start condition. <4> Set bit 6 (LREL0) of IICC0 to 1 before $\overline{\text{ACK}}$ is returned (4 to 80 clocks after setting IICE0 to 1), to forcibly disable detection.	p.591	<input type="checkbox"/>
			Setting transfer clock frequency	Determine the transfer clock frequency by using SMC0, CL01, CL00 (bits 3, 1, and 0 of IICL0), and CLX0 (bit 0 of IICX0) before enabling the operation (IICE0 = 1). To change the transfer clock frequency, clear IICE0 to 0 once.	p.591	<input type="checkbox"/>
			STT0, SPT0: Bits 1, 0 of IIC control register 0 (IICC0)	Setting STT0 and SPT0 (bits 1 and 0 of IICC0) again after they are set and before they are cleared to 0 is prohibited.	p.591	<input type="checkbox"/>
			Reserving transmission	When transmission is reserved, set SPIE0 (bit 4 of IICL0) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to IIC0 after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set SPIE0 to 1 when MST0 (bit 7 of IICS0) is detected by software.	p.591	<input type="checkbox"/>
			Chapter 16	Soft	DMA controller	DBCn: DMA byte count register n
If the general-purpose register is specified or the internal RAM space is exceeded as a result of continuous transfer, the general-purpose register or SFR space are written or read, resulting in loss of data in these spaces. Be sure to set the number of times of transfer that is within the internal RAM space.	p.634	<input type="checkbox"/>				

Chapter	Classification	Function	Details of Function	Caution	Page	
Chapter 16	Soft	DMA controller	DRCn: DMA operation control register n	The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENn flag is enabled only when DSTn = 0. When a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, therefore, set DSTn to 0 and then DENn to 0 (for details, refer to 16.5.5 Forcible termination by software).	p.637 <input type="checkbox"/>	
			Example of setting for holding DMA transfer pending by DWAITn	When DMA transfer is held pending while using both DMA channels, be sure to hold the DMA transfer pending for both channels (by setting DWAIT0 and DWAIT1 to 1). If the DMA transfer of one channel is executed while that of the other channel is held pending, DMA transfer might not be held pending for the latter channel.	p.650 <input type="checkbox"/>	
			Forced termination of DMA Transfer	In example 3, the system is not required to wait two clock cycles after DWAITn is set to 1. In addition, the system does not have to wait two clock cycles after clearing DSTn to 0, because more than two clock cycles elapse from when DSTn is cleared to 0 to when DENn is cleared to 0.	p.652 <input type="checkbox"/>	
			Priority	During DMA transfer, a request from the other DMA channel is held pending even if generated. The pending DMA transfer is started after the ongoing DMA transfer is completed. If two DMA requests are generated at the same time, however, DMA channel 0 takes priority over DMA channel 1. If a DMA request and an interrupt request are generated at the same time, the DMA transfer takes precedence, and then interrupt servicing is executed.	p.652 <input type="checkbox"/>	
	Hard			Response time	The response time of DMA transfer is as follows. (See Table 16-2.)	p.653 <input type="checkbox"/>
				Soft	Operation in standby mode	The DMA controller operates as follows in the standby mode. (See Table 16-3.)
		Soft		DMA pending instruction	Even if a DMA request is generated, DMA transfer is held pending immediately after the following instructions. <ul style="list-style-type: none"> <li>• CALL !addr16</li> <li>• CALL \$!addr20</li> <li>• CALL !!addr20</li> <li>• CALL rp</li> <li>• CALLT [addr5]</li> <li>• BRK</li> <li>• Bit manipulation instructions for registers IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H and PSW each.</li> </ul>	p.654 <input type="checkbox"/>
				Operation if address in general-purpose register area or other than those of internal RAM area is specified	The address indicated by DRA0n is incremented during DMA transfer. If the address is incremented to an address in the general-purpose register area or exceeds the area of the internal RAM, the following operation is performed. <ul style="list-style-type: none"> <li>● In mode of transfer from SFR to RAM The data of that address is lost.</li> <li>● In mode of transfer from RAM to SFR Undefined data is transferred to SFR.</li> </ul> In either case, malfunctioning may occur or damage may be done to the system. Therefore, make sure that the address is within the internal RAM area other than the general-purpose register area.	p.654 <input type="checkbox"/>
				Interrupt functions	IF0L, IF0H, IF1L, IF1H, IF2L, IF2H: Interrupt request flag registers	Be sure to clear bits 5 to 7 of IF2H to 0. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.



Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 17	Soft	Interrupt functions	IF0L, IF0H, IF1L, IF1H, IF2L, IF2H: Interrupt request flag registers	When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1). If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions. <pre> mov a, IF0L and a, #0FEH mov IF0L, a </pre> In this case, even if the request flag of another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.	p.664 <input type="checkbox"/>
			MK0L, MK0H, MK1L, MK1H, MK2L, MK2H: Interrupt mask flag registers	Be sure to set bits 5 to 7 of MK2H to 1.	p.665 <input type="checkbox"/>
			PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H: Priority specification flag registers	Be sure to set bits 5 to 7 of PR02H and PR12H to 1.	p.667 <input type="checkbox"/>
			EGP0, EGP1: External interrupt rising edge enable registers, EGN0, EGN1: External interrupt falling edge enable registers	Select the port mode by clearing EGPn and EGNn to 0 because an edge may be detected when the external interrupt function is switched to the port function.	p.669 <input type="checkbox"/>
			Software interrupt request acknowledgment	Do not use the RETI instruction for restoring from the software interrupt.	p.673 <input type="checkbox"/>
			BRK instruction	The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged.	p.677 <input type="checkbox"/>
Chapter 18	Soft	Key interrupt function	KRM: Key return mode register	If any of the KRM0 to KRM7 bits used is set to 1, set bits 0 to 7 (PU70 to PU77) of the corresponding pull-up resistor register 7 (PU7) to 1.	p.679 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page	
Chapter 18	Soft	Key interrupt function	KRM: Key return mode register	An interrupt will be generated if the target bit of the KRM register is set while a low level is being input to the key interrupt input pin. To ignore this interrupt, set the KRM register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input low-level width (250 ns or more).	p.679 <input type="checkbox"/>	
				The bits not used in the key interrupt mode can be used as normal ports.	p.679 <input type="checkbox"/>	
Chapter 19	Soft	Standby function	-	The STOP mode can be used only when the CPU is operating on the main system clock. The STOP mode cannot be set while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.	p.680 <input type="checkbox"/>	
				When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.	p.680 <input type="checkbox"/>	
				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 the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.	p.680 <input type="checkbox"/>	
				It can be selected by the option byte whether the internal low-speed oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 24 OPTION BYTE.	p.680 <input type="checkbox"/>	
	Hard			OSTC: Oscillation stabilization time counter status register	After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.	p.681 <input type="checkbox"/>
					The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTC. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows. • Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTC Note, therefore, that only the status up to the oscillation stabilization time set by OSTC is set to OSTC after STOP mode is released.	p.681 <input type="checkbox"/>
	Soft			OSTS: Oscillation stabilization time select register	The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).	p.681 <input type="checkbox"/>
					To set the STOP mode when the X1 clock is used as the CPU clock, set OSTC before executing the STOP instruction.	p.682 <input type="checkbox"/>
					Setting the oscillation stabilization time to 20 μs or less is prohibited.	p.682 <input type="checkbox"/>
					Before changing the setting of the OSTC register, confirm that the count operation of the OSTC register is completed.	p.682 <input type="checkbox"/>
Do not change the value of the OSTC register during the X1 clock oscillation stabilization time.					p.682 <input type="checkbox"/>	
The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTC. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows. • Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTC Note, therefore, that only the status up to the oscillation stabilization time set by OSTC is set to OSTC after STOP mode is released.					p.682 <input type="checkbox"/>	
Hard				The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).	p.682 <input type="checkbox"/>	

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 19	Soft	Standby function	STOP mode	Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.	p.688 <input type="checkbox"/>
				To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.	p.690 <input type="checkbox"/>
				To stop the internal low-speed oscillation clock in the STOP mode, use an option byte to stop the watchdog timer operation in the HALT/STOP mode (bit 0 (WDSTBYON) of 000C0H = 0), and then execute the STOP instruction.	p.690 <input type="checkbox"/>
				To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the internal high-speed oscillation clock before the execution of the STOP instruction. Before changing the CPU clock from the internal high-speed oscillation clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).	p.690 <input type="checkbox"/>
Chapter 20	Hard	Reset function	-	For an external reset, input a low level for 10 $\mu$ s or more to the $\overline{\text{RESET}}$ pin. (If an external reset is effected upon power application, the period during which the supply voltage is outside the operating range ( $V_{DD} < 1.8 \text{ V}$ ) is not counted in the 10 $\mu$ s. However, the low-level input may be continued before POC is released.)	p.695 <input type="checkbox"/>
				During reset input, the X1 clock, XT1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.	p.695 <input type="checkbox"/>
				When the STOP mode is released by a reset, the RAM contents in the STOP mode are held during reset input. However, because SFR and 2nd SFR are initialized, the port pins become high-impedance, except for P130, which is set to low-level output.	p.695 <input type="checkbox"/>
	Soft	Block diagram of reset function	An LVI circuit internal reset does not reset the LVI circuit.	p.696 <input type="checkbox"/>	
			Watchdog timer overflow	A watchdog timer internal reset resets the watchdog timer.	p.697 <input type="checkbox"/>
			RESF: Reset control flag register	Do not read data by a 1-bit memory manipulation instruction. When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, LVIRF flag may become 1 from the beginning depending on the power-on waveform.	p.703 <input type="checkbox"/> p.703 <input type="checkbox"/>
Chapter 21	Soft	Power-on-clear circuit	-	If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage ( $V_{DD}$ ) exceeds $2.07 \text{ V} \pm 0.2 \text{ V}$ .	pp.704, <input type="checkbox"/> 705
				If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.	
				Timing of generation of internal reset signal (LVIOFF = 1)	Set the low-voltage detector by software after the reset status is released (see CHAPTER 22 LOW-VOLTAGE DETECTOR).
Timing of generation of internal reset signal (LVIOFF = 0)	Set the low-voltage detector by software after the reset status is released (see CHAPTER 22 LOW-VOLTAGE DETECTOR).	p.707 <input type="checkbox"/>			

Chapter	Classification	Function	Details of Function	Caution	Page
Chapter 21	Soft	Power-on-clear circuit	Caution for power-on-clear circuit	In a system where the supply voltage ( $V_{DD}$ ) fluctuates for a certain period in the vicinity of the POC detection voltage ( $V_{POC}$ ), 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.	p.708 <input type="checkbox"/>
Chapter 22	Soft	Low-voltage detector	LVIM: Low-voltage detection register	To stop LVI, follow either of the procedures below. <ul style="list-style-type: none"> <li>When using 8-bit memory manipulation instruction: Write 00H to LVIM.</li> <li>When using 1-bit memory manipulation instruction: Clear LVION to 0.</li> </ul>	p.713 <input type="checkbox"/>
				Input voltage from external input pin (EXLVI) must be $EXLVI < V_{DD}$ .	p.713 <input type="checkbox"/>
	Hard			When LVI is used in interrupt mode ( $LVIMD = 0$ ) and $LVISEL$ is set to 0, an interrupt request signal (INTLVI) that disables LVI operation (clears LVION) when the supply voltage ( $V_{DD}$ ) is less than or equal to the detection voltage ( $V_{LVI}$ ) (if $LVISEL = 1$ , input voltage of external input pin (EXLVI) is less than or equal to the detection voltage ( $V_{EXLVI}$ )) is generated and LVIF may be set to 1.	p.713 <input type="checkbox"/>
				LVIS: Low-voltage detection level select register	Be sure to clear bits 4 to 7 to "0". Change the LVIS value with either of the following methods. <ul style="list-style-type: none"> <li>When changing the value after stopping LVI                              &lt;1&gt; Stop LVI (<math>LVION = 0</math>).                              &lt;2&gt; Change the LVIS register.                              &lt;3&gt; Set to the mode used as an interrupt (<math>LVIMD = 0</math>).                              &lt;4&gt; Mask LVI interrupts (<math>LVIMK = 1</math>).                              &lt;5&gt; Enable LVI operation (<math>LVION = 1</math>).                              &lt;6&gt; Before cancelling the LVI interrupt mask (<math>LVIMK = 0</math>), clear it with software because an LVIF flag may be set when LVI operation is enabled.</li> <li>When changing the value after setting to the mode used as an interrupt (<math>LVIMD = 0</math>)                              &lt;1&gt; Mask LVI interrupts (<math>LVIMK = 1</math>).                              &lt;2&gt; Set to the mode used as an interrupt (<math>LVIMD = 0</math>).                              &lt;3&gt; Change the LVIS register.                              &lt;4&gt; Before cancelling the LVI interrupt mask (<math>LVIMK = 0</math>), clear it with software because an LVIF flag may be set when the LVIS register is changed.</li> </ul>
	Soft			When an input voltage from the external input pin (EXLVI) is detected, the detection voltage ( $V_{EXLVI}$ ) is fixed. Therefore, setting of LVIS is not necessary.	p.715 <input type="checkbox"/>
				Used as reset (when detecting level of supply voltage ( $V_{DD}$ )) ( $LVIOFF = 1$ )	<1> must always be executed. When $LVIMK = 0$ , an interrupt may occur immediately after the processing in <4>. If supply voltage ( $V_{DD}$ ) $\geq$ detection voltage ( $V_{LVI}$ ) when $LVIMD$ is set to 1, an internal reset signal is not generated.
				Used as reset (when detecting level of supply voltage ( $V_{DD}$ )) ( $LVIOFF = 0$ )	Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows: <ul style="list-style-type: none"> <li>Does not perform low-voltage detection during <math>LVION = 0</math>.</li> <li>If a reset is generated while <math>LVION = 0</math>, <math>LVION</math> will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution.</li> </ul> This is due to the fact that while the pulse width detected by LVI must be 200 $\mu$ s max., $LVION = 1$ is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.

Chapter	Classification	Function	Details of Function	Caution	Page
Chapter 22	Soft	Low-voltage detector	Used as reset (when detecting level of input voltage from external input pin (EXLVI))	<1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.	p.721 <input type="checkbox"/>
				If input voltage from external input pin (EXLVI) $\geq$ detection voltage ( $V_{EXLVI} = 1.21 \text{ V (TYP.)}$ ) when LVIMD is set to 1, an internal reset signal is not generated.	p.721 <input type="checkbox"/>
				Input voltage from external input pin (EXLVI) must be $EXLVI < V_{DD}$ .	p.721 <input type="checkbox"/>
	Soft		Used as interrupt (when detecting level of supply voltage ( $V_{DD}$ )) (LVIOFF = 0)	<p>Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:</p> <ul style="list-style-type: none"> <li>Does not perform low-voltage detection during LVION = 0.</li> <li>If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution.</li> </ul> <p>This is due to the fact that while the pulse width detected by LVI must be 200 <math>\mu\text{s}</math> max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.</p>	p.725 <input type="checkbox"/>
	Hard		Used as interrupt (when detecting level of input voltage from external input pin (EXLVI))	The input voltage from the external input pin (EXLVI) must be $EXLVI < V_{DD}$ .	p.727 <input type="checkbox"/>
Soft		Caution for low-voltage detector	<p>In a system where the supply voltage (<math>V_{DD}</math>) fluctuates for a certain period in the vicinity of the LVI detection voltage (<math>V_{LVI}</math>), the operation is as follows depending on how the low-voltage detector is used.</p> <p>Operation example 1: When used as reset The system may be repeatedly reset and released from the reset status. The time from reset release through microcontroller operation start can be set arbitrarily by the following action.</p> <p>&lt;Action&gt; After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (see Figure 22-11).</p> <p>Operation example 2: When used as interrupt Interrupt requests may be generated frequently. Take the following action.</p> <p>&lt;Action&gt; Confirm that “supply voltage (<math>V_{DD}</math>) <math>\geq</math> detection voltage (<math>V_{LVI}</math>)” when detecting the falling edge of <math>V_{DD}</math>, or “supply voltage (<math>V_{DD}</math>) <math>&lt;</math> detection voltage (<math>V_{LVI}</math>)” when detecting the rising edge of <math>V_{DD}</math>, in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 1 (LVIF) of interrupt request flag register 0L (IF0L) to 0.</p> <p>For a system with a long supply voltage fluctuation period near the LVI detection voltage, take the above action after waiting for the supply voltage fluctuation time.</p>	pp.729 to 732 <input type="checkbox"/>	

Chapter	Classification	Function	Details of Function	Caution	Page	
Chapter 22	Hard	Low-voltage detector	Cautions for low-voltage detector	There is some delay from the time supply voltage ( $V_{DD}$ ) < LVI detection voltage ( $V_{LVI}$ ) until the time LVI reset has been generated.	p.732 <input type="checkbox"/>	
				In the same way, there is also some delay from the time LVI detection voltage ( $V_{LVI}$ ) $\leq$ supply voltage ( $V_{DD}$ ) until the time LVI reset has been released (see Figure 22-12). See the timing in Figure 21-2 (2) When LVI is ON upon power application (option byte: LVIOFF = 0) for the reset processing time until the normal operation is entered after the LVI reset is released.		
Chapter 23	Soft	Regulator	RMC: Regulator mode control register	The RMC register can be rewritten only in the low consumption current mode (refer to Table 23-1). In other words, rewrite this register during CPU operation with the subsystem clock ( $f_{XT}$ ) while the high-speed system clock ( $f_{MX}$ ) and internal high-speed oscillation clock ( $f_{IH}$ ) are both stopped.	p.733 <input type="checkbox"/>	
				When using the setting fixed to the low consumption current mode, the RMC register can be used in the following cases. <When X1 clock is selected as the CPU clock> $f_X \leq 5$ MHz and $f_{CLK} \leq 5$ MHz <When the internal high-speed oscillation clock, external input clock, or subsystem clock are selected for the CPU clock> $f_{CLK} \leq 5$ MHz	p.733 <input type="checkbox"/>	
				The self-programming function is disabled in the low consumption current mode.	p.733 <input type="checkbox"/>	
Chapter 24	Soft	Option byte	-	Be sure to set FFH to 000C2H (000C2H/010C2H when the boot swap operation is used).	p.735 <input type="checkbox"/>	
				000C0H/010C0H	Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.	p.735 <input type="checkbox"/>
				000C1H/010C1H	Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.	p.735 <input type="checkbox"/>
				000C2H/010C2H	Set FFH to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.	p.735 <input type="checkbox"/>
				000C3H/010C3H	Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.	p.736 <input type="checkbox"/>
				000C0H/010C0H	The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.	p.737 <input type="checkbox"/>
				000C1H/010C1H	Be sure to set bits 7 to 1 to "1".	p.737 <input type="checkbox"/>
					Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows: • Does not perform low-voltage detection during LVION = 0. • If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution. This is due to the fact that while the pulse width detected by LVI must be 200 $\mu$ s max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.	p.737 <input type="checkbox"/>
				000C3H/010C3H	Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value. Be sure to set 000010B to bits 6 to 1.	p.738 <input type="checkbox"/>
Setting of option byte	To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify the option byte to 010C0H to 010C3H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.	p.739 <input type="checkbox"/>				

Chapter	Classification	Function	Details of Function	Cautions	Page	
Chapter 25	Hard	Flash memory	Security settings	After the security setting for the batch erase is set, erasure cannot be performed for the device. In addition, even if a write command is executed, data different from that which has already been written to the flash memory cannot be written, because the erase command is disabled.	p.750 <input type="checkbox"/>	
			Flash memory programming by self-programming	The self-programming function cannot be used when the CPU operates with the subsystem clock. In the self-programming mode, call the self-programming start library (FlashStart). To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the self-programming library.	p.753 <input type="checkbox"/>	
	Soft				In the self-programming mode, call the self-programming start library (FlashStart).	p.753 <input type="checkbox"/>
					The self-programming function is disabled in the low consumption current mode. For details of the low consumption current mode, see CHAPTER 23 REGULATOR.	p.753 <input type="checkbox"/>
					Disable DMA operation (DENn = 0) during the execution of self programming library functions.	p.753 <input type="checkbox"/>
			Flash shield window function	If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.	p.757 <input type="checkbox"/>	
	Chapter 26	Hard	On-chip debug function	Connecting QB-MINI2 to 78K0R/KH3	The 78K0R/KH3 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. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.	p.758 <input type="checkbox"/>
				When communicating in 2-line mode, a clock with a frequency of half that of the CPU clock frequency is output from the TOOL1 pin. A resistor or ferrite bead can be used as a countermeasure against fluctuation of the power supply caused by that clock.	p.758 <input type="checkbox"/>	
Chapter 27	Soft	BCD correction circuit	Addition	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.	p.762 <input type="checkbox"/>	
			Subtraction	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.	p.763 <input type="checkbox"/>	
Chapter 28	Soft	Instruction set	PREFIX instruction	Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.	p.767 <input type="checkbox"/>	

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 29	Hard	Electrical specifications (standard products)	–	The 78K0R/KH3 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. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.	p.785 <input type="checkbox"/>
			Absolute maximum ratings	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.	pp.785, 786 <input type="checkbox"/>
			X1 oscillator characteristics	When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance. <ul style="list-style-type: none"> <li>• Keep the wiring length as short as possible.</li> <li>• Do not cross the wiring with the other signal lines.</li> <li>• Do not route the wiring near a signal line through which a high fluctuating current flows.</li> <li>• Always make the ground point of the oscillator capacitor the same potential as V<sub>SS</sub>.</li> <li>• Do not ground the capacitor to a ground pattern through which a high current flows.</li> <li>• Do not fetch signals from the oscillator.</li> </ul>	p.787 <input type="checkbox"/>
				Since the CPU is started by the internal high-speed oscillation clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.	p.787 <input type="checkbox"/>
			XT1 oscillator characteristics	When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance. <ul style="list-style-type: none"> <li>• Keep the wiring length as short as possible.</li> <li>• Do not cross the wiring with the other signal lines.</li> <li>• Do not route the wiring near a signal line through which a high fluctuating current flows.</li> <li>• Always make the ground point of the oscillator capacitor the same potential as V<sub>SS</sub>.</li> <li>• Do not ground the capacitor to a ground pattern through which a high current flows.</li> <li>• Do not fetch signals from the oscillator.</li> </ul>	p.789 <input type="checkbox"/>
				The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.	p.789 <input type="checkbox"/>



Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 29	Hard	Electrical specifications (standard products)	Recommended oscillator constants	<p>The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.</p> <p>When doing so, check the conditions for using the AMPH bit, RMC register, and whether to enter or exit the STOP mode.</p> <p>The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.</p>	pp.790 to 793 <input type="checkbox"/>
				<p>The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.</p> <p>When doing so, check the conditions for using the RMC register, and whether to enter or exit the STOP mode.</p> <p>The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.</p>	p.794 <input type="checkbox"/>
			DC characteristics	<p>P02 to P04, P43, P45, P95 to P97, P125 to P127, P142 to P144 do not output high level in N-ch open-drain mode.</p>	p.795 <input type="checkbox"/>
				<p>The maximum value of <math>V_{IH}</math> of pins P02 to P04, P43, P45, P95 to P97, P125 to P127, and P142 to P144 is <math>V_{DD}</math>, even in the N-ch open-drain mode.</p>	p.797 <input type="checkbox"/>
				<p>For P122/EXCLK, the value of <math>V_{IH}</math> differ according to the input port mode or external clock mode. Make sure to satisfy the DC characteristics of EXCLK in external clock input mode.</p>	p.797 <input type="checkbox"/>
				<p>For P122/EXCLK, the value of <math>V_{IL}</math> differ according to the input port mode or external clock mode. Make sure to satisfy the DC characteristics of EXCLK in external clock input mode.</p>	pp.797, 798 <input type="checkbox"/>
			External bus interface	<p>CLKOUT output is not used during CLKOUT asynchronous operation, but a CPU wait occurs according to the setting of bits 4 and 5 (EW0, EW1) of the memory expansion mode control register (MEM). When <math>f_{CLK}</math> is sufficiently high, insert a wait by setting the EW0 and EW1 bits.</p>	p.816 <input type="checkbox"/>
				<p>Do not use the WAIT pin during CLKOUT asynchronous operation.</p> <p>Use the separate bus mode during CLKOUT asynchronous operation.</p>	p.816 <input type="checkbox"/>
			During communication at same potential (UART mode) (dedicated baud rate generator output)	<p>Select the normal input buffer for RxDi and the normal output mode for TxDi by using the PIMg and POMg registers.</p>	p.818 <input type="checkbox"/>
			During communication at same potential (CSI mode) (master mode, SCKp... internal clock output)	<p>Select the normal input buffer for Slj and the normal output mode for SOj and SCKj by using the PIMg and POMg registers.</p>	p.819 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 29	Hard	Electrical specifications (standard products)	During communication at same potential (CSI mode) (slave mode, $\overline{\text{SCKp}}$ ... external clock input)	Select the normal input buffer for $\text{Slj}$ and $\overline{\text{SCKj}}$ and the normal output mode for $\text{SOj}$ by using the PIMg and POMg registers.	p.820 <input type="checkbox"/>
			During communication at same potential (simplified I <sup>2</sup> C mode)	Select the normal input buffer and the N-ch open-drain output ( $V_{\text{DD}}$ tolerance) mode for $\text{SDAr}$ and the normal output mode for $\text{SCLr}$ by using the PIMg and POMg registers.	p.823 <input type="checkbox"/>
			During communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output)	Select the TTL input buffer for $\text{RxDq}$ and the N-ch open-drain output ( $V_{\text{DD}}$ tolerance) mode for $\text{TxDq}$ by using the PIMg and POMg registers.	pp.824, 825, 827 <input type="checkbox"/>
			During communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{\text{SCKp}}$ ... internal clock output)	Select the TTL input buffer for $\text{Slp}$ and the N-ch open-drain output ( $V_{\text{DD}}$ tolerance) mode for $\text{SOp}$ and $\overline{\text{SCKp}}$ by using the PIMg and POMg registers.	pp.828 to 830 <input type="checkbox"/>
			During communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, $\overline{\text{SCKp}}$ ... external clock input)	Select the TTL input buffer for $\text{Slp}$ and $\overline{\text{SCKp}}$ and the N-ch open-drain output ( $V_{\text{DD}}$ tolerance) mode for $\text{SOp}$ by using the PIMg and POMg registers.	pp.832, 833 <input type="checkbox"/>
			During communication at different potential (2.5 V, 3 V) (simplified I <sup>2</sup> C mode)	Select the TTL input buffer and the N-ch open-drain output ( $V_{\text{DD}}$ tolerance) mode for $\text{SDAr}$ and the N-ch open-drain output ( $V_{\text{DD}}$ tolerance) mode for $\text{SCLr}$ by using the PIMg and POMg registers.	pp.834, 835 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 30	Hard	Electrical specifications ((A) grade products)	–	The 78K0R/KH3 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. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.	p.845 <input type="checkbox"/>
			Absolute maximum ratings	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.	pp.845, 846 <input type="checkbox"/>
			X1 oscillator characteristics	When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance. <ul style="list-style-type: none"> <li>• Keep the wiring length as short as possible.</li> <li>• Do not cross the wiring with the other signal lines.</li> <li>• Do not route the wiring near a signal line through which a high fluctuating current flows.</li> <li>• Always make the ground point of the oscillator capacitor the same potential as V<sub>SS</sub>.</li> <li>• Do not ground the capacitor to a ground pattern through which a high current flows.</li> <li>• Do not fetch signals from the oscillator.</li> </ul>	p.847 <input type="checkbox"/>
				Since the CPU is started by the internal high-speed oscillation clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.	p.847 <input type="checkbox"/>
				When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance. <ul style="list-style-type: none"> <li>• Keep the wiring length as short as possible.</li> <li>• Do not cross the wiring with the other signal lines.</li> <li>• Do not route the wiring near a signal line through which a high fluctuating current flows.</li> <li>• Always make the ground point of the oscillator capacitor the same potential as V<sub>SS</sub>.</li> <li>• Do not ground the capacitor to a ground pattern through which a high current flows.</li> <li>• Do not fetch signals from the oscillator.</li> </ul>	p.849 <input type="checkbox"/>
				The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.	p.849 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 30	Hard	Electrical specifications ((A) grade products)	Recommended oscillator constants	The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit. When doing so, check the conditions for using the AMPH bit, RMC register, and whether to enter or exit the STOP mode. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.	pp.850 to 853 <input type="checkbox"/>
				The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit. When doing so, check the conditions for using the RMC register, and whether to enter or exit the STOP mode. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KH3 so that the internal operation conditions are within the specifications of the DC and AC characteristics.	p.854 <input type="checkbox"/>
			DC characteristics	P02 to P04, P43, P45, P95 to P97, P125 to P127, P142 to P144 do not output high level in N-ch open-drain mode.	p.855 <input type="checkbox"/>
				The maximum value of $V_{IH}$ of pins P02 to P04, P43, P45, P95 to P97, P125 to P127, and P142 to P144 is $V_{DD}$ , even in the N-ch open-drain mode.	p.857 <input type="checkbox"/>
				For P122/EXCLK, the value of $V_{IH}$ differ according to the input port mode or external clock mode. Make sure to satisfy the DC characteristics of EXCLK in external clock input mode.	p.857 <input type="checkbox"/>
	Soft	During communication at same potential (UART mode) (dedicated baud rate generator output)	External bus interface	CLKOUT output is not used during CLKOUT asynchronous operation, but a CPU wait occurs according to the setting of bits 4 and 5 (EW0, EW1) of the memory expansion mode control register (MEM). When $f_{CLK}$ is sufficiently high, insert a wait by setting the EW0 and EW1 bits.	p.875 <input type="checkbox"/>
				Do not use the $\overline{WAIT}$ pin during CLKOUT asynchronous operation. Use the separate bus mode during CLKOUT asynchronous operation.	p.875 <input type="checkbox"/>
			During communication at same potential (UART mode) (dedicated baud rate generator output)	Select the normal input buffer for RxDi and the normal output mode for TxDi by using the PIMg and POMg registers.	p.877 <input type="checkbox"/>
			During communication at same potential (CSI mode) (master mode, SCKp... internal clock input)	Select the normal input buffer for SIj and the normal output mode for SOj and $\overline{SCKj}$ by using the PIMg and POMg registers.	p.878 <input type="checkbox"/>

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 30	Soft	Electrical specifications ((A) grade products)	During communication at same potential (CSI mode) (slave mode, $\overline{\text{SCKp}}$ ... external clock input)	Select the normal input buffer for $\text{Slj}$ and $\overline{\text{SCKj}}$ and the normal output mode for $\text{SOj}$ by using the PIMg and POMg registers.	p.879 <input type="checkbox"/>
			During communication at same potential (simplified I <sup>2</sup> C mode)	Select the normal input buffer and the N-ch open-drain output ( $V_{\text{DD}}$ tolerance) mode for $\text{SDAr}$ and the normal output mode for $\text{SCLr}$ by using the PIMg and POMg registers.	p.882 <input type="checkbox"/>
			During communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output)	Select the TTL input buffer for $\text{RxDq}$ and the N-ch open-drain output ( $V_{\text{DD}}$ tolerance) mode for $\text{TxDq}$ by using the PIMg and POMg registers.	pp.883, 884, 886 <input type="checkbox"/>
			During communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{\text{SCKp}}$ ... internal clock output)	Select the TTL input buffer for $\text{Slp}$ and the N-ch open-drain output ( $V_{\text{DD}}$ tolerance) mode for $\text{SOp}$ and $\overline{\text{SCKp}}$ by using the PIMg and POMg registers.	pp.887 to 889 <input type="checkbox"/>
			During communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, $\overline{\text{SCKp}}$ ... external clock input)	Select the TTL input buffer for $\text{Slp}$ and $\overline{\text{SCKp}}$ and the N-ch open-drain output ( $V_{\text{DD}}$ tolerance) mode for $\text{SOp}$ by using the PIMg and POMg registers.	pp.891, 892 <input type="checkbox"/>
			During communication at different potential (2.5 V, 3 V) (simplified I <sup>2</sup> C mode)	Select the TTL input buffer and the N-ch open-drain output ( $V_{\text{DD}}$ tolerance) mode for $\text{SDAr}$ and the N-ch open-drain output ( $V_{\text{DD}}$ tolerance) mode for $\text{SCLr}$ by using the PIMg and POMg registers.	pp.893, 894 <input type="checkbox"/>
			Chapter 32	Hard	Recommended soldering conditions
	Do not use different soldering methods together (except for partial heating).	p.904 <input type="checkbox"/>			

## APPENDIX C REVISION HISTORY

### C.1 Major Revisions in This Edition

(1/5)

Page	Description	Classification
<b>Throughout</b>		
–	Change of status of <b>(A) grade products of the expanded-specification products</b> from under development to mass production	(b)
<b>CHAPTER 1 OUTLINE</b>		
p.18	Change of <b>1.1 Differences Between Conventional-Specification Products (<math>\mu</math>PD78F117x) and Expanded-Specification Products (<math>\mu</math>PD78F117xA)</b>	(c)
<b>CHAPTER 3 CPU ARCHITECTURE</b>		
pp.68 to 72	Change of <b>Figure 3-7 to Figure 3-11 Correspondence Between Data Memory and Addressing</b>	(c)
p.74	Addition of <b>Caution to 3.2.1 (3) Stack pointer (SP)</b>	(c)
<b>CHAPTER 4 PORT FUNCTIONS</b>		
p.148	Change of <b>Figure 4-36. Block Diagram of P110 and P111</b>	(c)
<b>CHAPTER 5 EXTERNAL BUS INTERFACE</b>		
p.190	Addition of <b>5.6 Number of Instructed Wait Cycles According to External Wait Cycles</b>	(c)
<b>CHAPTER 6 CLOCK GENERATOR</b>		
pp.204, 205	Addition of <b>f<sub>MAINC</sub> to Figure 6-1. Block Diagram of Clock Generator and Remark</b>	(c)
p.206	Change of description of AMPH bit in <b>Figure 6-2. Format of Clock Operation Mode Control Register (CMC)</b>	(c)
p.216	Change of <b>Caution 5 in Figure 6-8. Format of Operation Speed Mode Control Register (OSMC)</b> and addition of <b>Caution 4</b>	(c)
p.236	Change of description of AMPH bit in <b>Table 6-4. CPU Clock Transition and SFR Register Setting Examples (1/4) (2)</b> and addition of <b>Remark</b>	(c)
p.237	Change of description of AMPH bit in <b>Table 6-4. CPU Clock Transition and SFR Register Setting Examples (2/4) (4)</b> and addition of <b>Remark</b>	(c)
p.239	Change of <b>(9) CPU clock changing from subsystem clock (D) to high-speed system clock (C) in Table 6-4. CPU Clock Transition and SFR Register Setting Examples (4/4)</b>	(c)
p.239	Change of <b>(11) • STOP mode (H) set while CPU is operating with internal high-speed oscillation clock (B) • STOP mode (I) set while CPU is operating with high-speed system clock (C) in Table 6-4. CPU Clock Transition and SFR Register Setting Examples (4/4)</b>	(c)
p.242	Change of <b>Table 6-6. Maximum Time Required for Main System Clock Switchover</b>	(c)
p.242	Change of <b>Table 6-8. Maximum Number of Clocks Required in Type 2</b>	(c)
p.243	Change of <b>Table 6-9. Maximum Number of Clocks Required in Type 3</b> and addition of <b>Remark</b>	(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

Page	Description	Classification
<b>CHAPTER 7 TIMER ARRAY UNIT</b>		
p.247	Change of <b>Figure 7-1. Block Diagram of Timer Array Unit</b>	(c)
p.264	Change of <b>Figure 7-14. Start Timing (In One-count Mode)</b>	(a)
p.265	Change of <b>Figure 7-15. Start Timing (In Capture &amp; One-count Mode)</b>	(a)
p.272	Change of description of ISC1 and ISC0 bits in <b>Figure 7-22. Format of Input Switch Control Register (ISC)</b>	(a)
<b>CHAPTER 8 REAL-TIME COUNTER</b>		
p.331	Change of <b>Table 8-1. Configuration of Real-Time Counter</b>	(c)
p.333	Change of <b>8.3 Registers Controlling Real-Time Counter</b>	(c)
p.335	Change of description of AMPM bit in <b>Figure 8-3. Format of Real-Time Counter Control Register 0 (RTCC0)</b>	(c)
p.340	Change of description of <b>8.3 (7) Minute count register (MIN)</b>	(c)
p.340	Change of description of <b>8.3 (8) Hour count register (HOUR)</b>	(c)
p.345	Addition of description of DEV bit to <b>Figure 8-14. Format of Watch Error Correction Register (SUBCUD)</b>	(c)
p.347	Addition of <b>8.3 (17) Port mode registers 1, 3 (PM1, PM3)</b>	(c)
p.348	Change of <b>Figure 8-19. Procedure for Starting Operation of Real-Time Counter</b> and addition of <b>Note</b>	(c)
p.353	Addition of <b>Caution</b> to <b>8.4.5 1 Hz output of real-time counter</b>	(c)
p.353	Change of <b>8.4.6 32.768 kHz output of real-time counter</b>	(c)
p.353	Change of <b>8.4.7 512 Hz, 16.384 kHz output of real-time counter</b>	(c)
<b>CHAPTER 10 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER</b>		
p.369	Change of <b>Remark</b> in <b>10.4.1 Operation as output pin</b>	(c)
p.369	Change of <b>Figure 10-4. Remote Control Output Application Example</b>	(c)
<b>CHAPTER 11 A/D CONVERTER</b>		
p.374	Change of <b>Table 11-2. Settings of ADCS and ADCE</b>	(c)
p.374	Change of <b>Figure 11-5. Timing Chart When A/D voltage Comparator Is Used</b>	(c)
p.397	Change of <b>11.7 Cautions for A/D Converter (2) Reducing current when A/D converter is stopped</b>	(c)
p.401	Addition of <b>11.7 (13) Starting the A/D converter</b>	(c)
<b>CHAPTER 12 D/A CONVERTER</b>		
p.409	Change of <b>12.4.3 Cautions (1)</b>	(c)
p.409	Change of <b>12.4.3 Cautions (7)</b>	(c)
<b>CHAPTER 13 SERIAL ARRAY UNIT</b>		
p.422	Change of MDmn0 bit in <b>Figure 13-6. Format of Serial Mode Register mn (SMRmn) (2/2)</b>	(c)
p.424	Addition of <b>Note</b> to <b>Figure 13-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/3)</b>	(c)
p.426	Addition of <b>Caution</b> to <b>Figure 13-8. Format of Serial Data Register mn (SDRmn)</b>	(c)
p.436	Change of description of <b>Figure 13-17. Format of Input Switch Control Register (ISC)</b>	(a)
p.454	Change of interrupt in <b>13.5.2 Master reception</b>	(c)
p.455	Change of <b>Figure 13-32. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)</b>	(c)
p.457	Change of <b>Figure 13-35. Procedure for Resuming Master Reception</b>	(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

Page	Description	Classification
<b>CHAPTER 13 SERIAL ARRAY UNIT (continuation)</b>		
p.459	Change of <b>Figure 13-37. Flowchart of Master Reception (in Single-Reception Mode)</b>	(c)
p.460	Addition of <b>Figure 13-38. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)</b>	(c)
p.461	Addition of <b>Figure 13-39. Flowchart of Master Reception (in Continuous Reception Mode)</b>	(c)
p.474	Change of <b>Figure 13-51. Procedure for Resuming Slave Transmission</b>	(b)
p.476	Change of <b>Figure 13-53. Flowchart of Slave Transmission (in Single-Transmission Mode)</b>	(c)
p.478	Change of <b>Figure 13-55. Flowchart of Slave Transmission (in Continuous Transmission Mode)</b>	(c)
p.480	Change of <b>Figure 13-56. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)</b>	(c)
p.482	Change of <b>Figure 13-59. Procedure for Resuming Slave Reception</b>	(c)
p.484	Change of <b>Figure 13-61. Flowchart of Slave Reception (in Single-Reception Mode)</b>	(c)
p.486	Addition of <b>Caution</b> to <b>Figure 13-62. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)</b>	(c)
p.487	Addition of <b>Caution</b> to <b>Figure 13-63. Initial Setting Procedure for Slave Transmission/Reception</b>	(c)
p.489	Change of <b>Figure 13-65. Procedure for Resuming Slave Transmission/Reception</b> and addition of <b>Caution</b>	(c)
p.491	Change of <b>Figure 13-67. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)</b> and addition of <b>Caution</b>	(c)
p.493	Change of <b>Figure 13-69. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)</b> and addition of <b>Caution</b>	(c)
p.509	Change of <b>Figure 13-79. Example of Contents of Registers for UART Reception of UART (UART0, UART1, UART2, UART3) (1/2)</b>	(c)
p.512	Change of <b>Figure 13-82. Procedure for Resuming UART Reception</b>	(c)
p.514	Change of <b>Figure 13-84. Flowchart of UART Reception</b>	(c)
p.528	Change of <b>13.7 Operation of Simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21) Communication</b>	(c)
p.529	Change of transfer rate in <b>13.7.1 Address field transmission</b>	(b)
p.534	Change of transfer rate in <b>13.7.2 Data transmission</b>	(b)
p.537	Change of error detection flag and transfer rate in <b>13.7.3 Data reception</b>	(b)
p.542	Addition of <b>Caution</b> to <b>13.7.5 Calculating transfer rate</b>	(c)
p.542	Change of <b>Remark</b> in <b>13.7.5 Calculating transfer rate</b>	(c)
p.545	Addition of <b>Figure 13-105. Processing Procedure in Case of Parity Error or Overrun Error</b>	(c)
<b>CHAPTER 14 SERIAL INTERFACE IIC0</b>		
p.564	Change of description of STT0 bit in <b>Figure 14-6. Format of IIC Control Register 0 (IIC0) (3/4)</b>	(c)
<b>CHAPTER 16 DMA CONTROLLER</b>		
p.635	Addition of <b>Note</b> to <b>Figure 16-4. Format of DMA Mode Control Register n (DMCn) (1/2)</b>	(c)
p.640	Change of description in <b>16.5.1 CSI consecutive transmission</b>	(c)
p.641	Change of description in <b>Figure 16-7. Setting Example of CSI Consecutive Transmission</b>	(c)
pp.642, 643	Addition of <b>16.5.2 CSI master reception</b>	(c)
pp.644, 645	Addition of <b>16.5.3 CSI transmission/reception</b>	(c)
p.650	Change of description in <b>16.5.6 Holding DMA transfer pending by DWAITn</b>	(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



Page	Description	Classification
<b>CHAPTER 16 DMA CONTROLLER (continuation)</b>		
p.650	Addition of <b>Caution</b> to <b>Figure 16-12. Example of Setting for Holding DMA Transfer Pending by DWAITn</b>	(c)
pp.651, 652	Change of <b>16.5.7 Forced termination by software</b>	(c)
p.652	Change of <b>(1) Priority of DMA</b> in <b>16.6 Cautions on Using DMA Controller</b>	(c)
p.653	Change of <b>(2) DMA response time</b> in <b>16.6 Cautions on Using DMA Controller</b>	(c)
p.654	Change of description in <b>(4) DMA pending instruction</b> in <b>16.6 Cautions on Using DMA Controller</b>	(c)
<b>CHAPTER 17 INTERRUPT FUNCTIONS</b>		
p.659	Change of <b>(B) External maskable interrupt (INTPn)</b> in <b>Figure 17-1. Basic Configuration of Interrupt Function</b>	(c)
p.660	Addition of <b>(C) External maskable interrupt (INTKR)</b> to <b>Figure 17-1. Basic Configuration of Interrupt Function</b>	(c)
p.677	Addition of <b>instruction</b> to <b>17.4.4 Interrupt request hold</b>	(c)
<b>CHAPTER 18 KEY INTERRUPT FUNCTION</b>		
p.678	Change of <b>Table 18-2. Configuration of Key Interrupt</b>	(c)
p.679	Addition of <b>18.3 (2) Port mode register 7 (PM7)</b>	(c)
<b>CHAPTER 27 BCD CORRECTION CIRCUIT</b>		
p.762	Change of <b>27.3 BCD Correction Circuit Operation</b>	(a)
<b>CHAPTER 28 INSTRUCTION SET</b>		
p.767	Change of description in <b>28.1.4 PREFIX instruction</b>	(c)
p.783	Change of <b>Clocks of BT Mnemonic</b> in <b>Table 28-5. Operation List (16/17)</b>	(c)
p.784	Change of <b>Clocks of BF Mnemonic</b> in <b>Table 28-5. Operation List (17/17)</b>	(c)
<b>CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)</b>		
p.787	Deletion of <b>Remark</b> in <b>X1 Oscillator Characteristics</b>	(a)
p.789	Deletion of <b>Remark</b> in <b>XT1 Oscillator Characteristics</b>	(a)
pp.790, 792, 794	Change of <b>Caution</b> in <b>Recommended Oscillator Constants</b>	(c)
pp.791, 793	Addition of KYOCERA KINSEKI Co., Ltd. to <b>Recommended Oscillator Constants</b>	(c)
pp.802 to 806	Addition of <b>Remark</b> to <b>Supply current in DC Characteristics</b>	(c)
p.819	Change of <b>(b) During communication at same potential (CSI mode) (master mode, SCKp... internal clock output)</b> in <b>Serial interface: Serial array unit</b>	(b)
p.820	Change of <b>(c) During communication at same potential (CSI mode) (slave mode, SCKp... external clock input)</b> in <b>Serial interface: Serial array unit</b>	(b)
p.822	Addition of <b>Note</b> to <b>(d) During communication at same potential (simplified I<sup>2</sup>C mode)</b> in <b>Serial interface: Serial array unit</b>	(c)
pp.828, 829	Change of <b>(f) During communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output)</b> in <b>Serial interface: Serial array unit</b>	(b)
p.831	Change of <b>(g) During communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, SCKp... external clock input)</b> in <b>Serial interface: Serial array unit</b>	(b)
p.834	Addition of <b>Note</b> to <b>(h) During communication at different potential (2.5 V, 3 V) (simplified I<sup>2</sup>C mode)</b> in <b>Serial interface: Serial array unit</b>	(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

Page	Description	Classification
<b>CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) (continuation)</b>		
p.844	Change of Number of rewrites of <b>Expanded-specification products in Flash Memory Programming Characteristics</b>	(c)
<b>CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)</b>		
Throughout	Deletion of <b>(TARGET)</b>	(d)
p.847	Deletion of <b>Remark</b> in <b>X1 Oscillator Characteristics</b>	(a)
p.849	Deletion of <b>Remark</b> in <b>XT1 Oscillator Characteristics</b>	(a)
pp.850, 852, 854	Change of <b>Caution</b> in <b>Recommended Oscillator Constants</b>	(c)
pp.851, 853	Addition of KYOCERA KINSEKI Co., Ltd. to <b>Recommended Oscillator Constants</b>	(c)
pp.862 to 866	Addition of <b>Remark</b> to Supply current in <b>DC Characteristics</b>	(c)
p.878	Change of <b>(b) During communication at same potential (CSI mode) (master mode, SCKp... internal clock output)</b> in <b>Serial interface: Serial array unit</b>	(b)
p.879	Change of <b>(c) During communication at same potential (CSI mode) (slave mode, SCKp... external clock input)</b> in <b>Serial interface: Serial array unit</b>	(b)
p.881	Addition of <b>Note</b> to <b>(d) During communication at same potential (simplified I<sup>2</sup>C mode) in Serial interface: Serial array unit</b>	(c)
pp.887, 888	Change of <b>(f) During communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output)</b> in <b>Serial interface: Serial array unit</b>	(b)
p.890	Change of <b>(g) During communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, SCKp... external clock input)</b> in <b>Serial interface: Serial array unit</b>	(b)
p.893	Addition of <b>Note</b> to <b>(h) During communication at different potential (2.5 V, 3 V) (simplified I<sup>2</sup>C mode) in Serial interface: Serial array unit</b>	(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

## C.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/13)

Edition	Description	Chapter
2nd edition	Deletion of target from the capacitance value of the capacitor connected to the REGC pin	Throughout
	Change of corresponding pins of EV <sub>DD</sub> and V <sub>DD</sub> in <b>Table 2-1. Pin I/O Buffer Power Supplies</b>	CHAPTER 2 PIN FUNCTIONS
	Change of description in <b>2.2.21 REGC</b>	
	Change of description in <b>2.2.24 FLMD0</b>	
	Modification of P60 to P64, P110 and P111 in <b>Table 2-2. Connection of Unused Pins</b>	
	Modification of 12-D to 12-G, 37-A to 37-B, and 39 to 2-W in <b>Table 2-2. Connection of Unused Pins</b>	
	Modification of 12-D to 12-G, 37-A to 37-B, and 39 to 2-W in <b>Figure 2-1. Pin I/O Circuit List</b>	
	Change of address in <b>Figure 3-16. Configuration of General-Purpose Registers</b>	
	Addition of register and <b>Note</b> in <b>Table 3-5. SFR List</b>	
	Addition of the BCDADJ register to <b>Table 3-6. Extended SFR (2<sup>nd</sup> SFR) List (1/6)</b>	
	Change of address of the SOL0 register in <b>Table 3-6. Extended SFR (2<sup>nd</sup> SFR) List (2/6)</b>	
	Change of address and symbol of the SOL1 register in <b>Table 3-6. Extended SFR (2<sup>nd</sup> SFR) List (3/6)</b>	
	Addition of PIM register and POM register in block diagram	CHAPTER 4 PORT FUNCTIONS
	Change of corresponding pins of EV <sub>DD</sub> and V <sub>DD</sub> in <b>Table 4-1. Pin I/O Buffer Power Supplies</b>	
	Change of <b>Cautions 1</b> and <b>Cautions 2</b> in <b>4.2.1 Port 0</b>	
	Change of <b>Cautions 1</b> , <b>Cautions 2</b> , and <b>Cautions 3</b> in <b>4.2.2 Port 1</b>	
	Change of <b>Cautions 1</b> and addition of <b>Cautions 2</b> in <b>4.2.4 Port 3</b>	
	Change of <b>Cautions 2</b> and <b>Cautions 3</b> in <b>4.2.5 Port 4</b>	
	Addition of <b>Caution</b> to <b>4.2.7 Port 6</b>	
	Change of <b>Caution</b> in <b>4.2.10 Port 9</b>	
	Change of <b>Figure 4-36. Block Diagram of P110 and P111</b>	
	Addition of description to <b>4.2.12 Port 12</b> and change of <b>Cautions 2</b>	
	Change of <b>Caution</b> in <b>4.2.13 Port 13</b>	
	Change of <b>Cautions 1</b> and <b>Cautions 2</b> and addition of <b>Cautions 3</b> to <b>4.2.14 Port 14</b>	
	Change of <b>Caution</b> in <b>4.2.16 Port 16</b>	
	Addition description to <b>(4) Port input mode registers (PIM0, PIM4, PIM9, PIM12, PIM14)</b> and <b>(5) Port output mode registers (POM0, POM4, POM9, POM12, POM14)</b> in <b>4.3</b>	
	Change of <b>Figure 4-56. Bit Manipulation Instruction (P10)</b>	
	Addition of description to <b>5.1 Functions of External Bus Interface</b>	
	Addition of description to title of <b>(d)</b> in <b>Figure 5-5. Timing to Write to External Memory</b>	
	Change of <b>(b)</b> and <b>(d)</b> in <b>Figure 5-6. Timing to Read External Memory</b>	
	Addition of description to title of <b>(c)</b> in <b>Figure 5-7. Timing to Write to External Memory</b>	

Edition	Description	Chapter
2nd edition	Addition of <b>Notes 3</b> and change of <b>Cautions 2</b> in <b>Figure 6-6. Format of System Clock Control Register (CKC)</b>	CHAPTER 6 CLOCK GENERATOR
	Addition of <b>Cautions 5</b> to <b>Figure 6-8. Format of Operation Speed Mode Control Register (OSMC)</b>	
	Change of description in <b>6.3 (8) Internal high-speed oscillator trimming register (HIOTRM)</b> and addition of <b>Caution</b>	
	Change of <b>Figure 6-9. Format of Internal High-Speed Oscillator Trimming Register (HIOTRM)</b> and addition of <b>Caution</b>	
	Change of <b>Figure 6-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1))</b>	
2nd edition	Change of <b>Table 7-1. Configuration of Timer Array Unit 0</b>	CHAPTER 7 TIMER ARRAY UNIT
	Change of <b>Table 7-2. Configuration of Timer Array Unit 1</b>	
	Addition of <b>Note</b> to <b>Figure 7-6. Format of Timer Clock Select Register m (TPSm)</b>	
	Change of description of MASTERmn bit in <b>Figure 7-7. Format of Timer Mode Register mn (TMRmn) (1/3)</b>	
	Change of <b>Table 7-3. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode</b> and addition of <b>Remark</b>	
	Addition of <b>Caution</b> to <b>Figure 7-17. Format of Timer Input Select Register m (TISm)</b>	
	Addition of description to <b>7.3 (10) Timer output register m (TOm)</b>	
	Addition of description to <b>7.3 (12) Timer output mode register m (TOMm)</b>	
	Change of <b>Remark</b> in <b>Figure 7-22. Format of Input Switch Control Register (ISC)</b>	
	Change of description in <b>7.3 (14) Noise filter enable register 1, 2 (NFEN1, NFEN2)</b>	
	Change of <b>7.5.1 Timn edge detection circuit</b>	
	2nd edition	
Change of <b>Caution</b> in <b>Figure 8-2. Format of Peripheral Enable Register 0 (PER0)</b>		
Addition of description to <b>8.3 (15) Alarm hour register (ALARMWH)</b>		
Addition of <b>Note</b> to <b>Figure 8-18. Procedure for Starting Operation of Real-Time Counter</b>		
2nd edition	Change of <b>Cautions 1</b> and <b>Cautions 2</b> in <b>9.3 (1) Watchdog timer enable register (WDTE)</b>	CHAPTER 9 WATCHDOG TIMER
	Addition of <b>Caution 3</b> to <b>Table 9-4. Setting Window Open Period of Watchdog Timer</b>	
2nd edition	Change of <b>Figure 13-1. Block Diagram of Serial Array Unit 0</b>	CHAPTER 13 SERIAL ARRAY UNIT
	Change of <b>Figure 13-2. Block Diagram of Serial Array Unit 1</b>	
	Addition of settings and <b>Note</b> to <b>Figure 13-5. Format of Serial Clock Select Register m (SPSm)</b>	
	Addition of <b>Note</b> to <b>Figure 13-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/3)</b>	
	Change of <b>Figure 13-14. Format of Serial Output Enable Register m (SOEm)</b>	
	Addition of description to <b>13.3 (12) Serial output register m (SOM)</b>	
	Change of <b>Figure 13-15. Format of Serial Output Register m (SOM)</b>	
	Addition of <b>13.4 Operation stop mode</b>	
	Addition of <b>Note</b> to transfer rate	
	Change of description to <b>(a) Serial output register m (SOM)</b>	
	Change of <b>Figure 13-27. Procedure for Resuming Master Transmission</b>	

Edition	Description	Chapter
2nd edition	Change of <b>Figure 13-36. Timing Chart of Master Reception (in Single-Reception Mode)</b>	CHAPTER 13 SERIAL ARRAY UNIT
	Change of <b>Figure 13-41. Procedure for Resuming Master Transmission/Reception</b>	
	Change of <b>Figure 13-42. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode)</b>	
	Change of <b>Figure 13-44. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)</b>	
	Change of <b>Figure 13-45. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)</b>	
	Change of <b>Figure 13-49. Procedure for Resuming Slave Transmission</b>	
	Change of <b>Figure 13-50. Timing Chart of Slave Transmission (in Single-Transmission Mode)</b>	
	Change of <b>Figure 13-57. Procedure for Resuming Slave Reception</b>	
	Change of <b>Figure 13-58. Timing Chart of Slave Reception (in Single-Reception Mode)</b>	
	Change of <b>Figure 13-63. Procedure for Resuming Slave Transmission/Reception</b>	
	Change of <b>Figure 13-64. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)</b>	
	Change of <b>Figure 13-66. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)</b>	
	Change of <b>Figure 13-67. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)</b>	
	Change of <b>Note in 13.5.7 (2)</b>	
	Addition of setting and <b>Note</b> to <b>Table 13-2. Operating Clock Selection</b>	
	Change of transfer rate and addition of <b>Note</b>	
	Change of Transfer data length in <b>13.6.2 UART reception</b>	
	Change of setting of <b>(b) Serial output enable register m (SOEm)</b> and <b>(e) Serial mode register mr (SMRmr)</b> in <b>Figure 13-76. Example of Contents of Registers for UART Reception of UART (UART0, UART1, UART2, UART3)</b>	
	Change of <b>Figure 13-79. Timing Chart of UART Reception</b>	
	Change of Transfer data length in <b>13.6.3 LIN transmission</b>	
	Change of Transfer data length in <b>13.6.4 LIN reception</b>	
	Addition of setting and <b>Note</b> to <b>Table 13-3. Operating Clock Selection</b>	
	Change of <b>Figure 13-89. Initial Setting Procedure for Address Field Transmission</b>	
	Change of <b>Figure 13-90. Timing Chart of Address Field Transmission</b>	
	Change of <b>Figure 13-91. Flowchart of Address Field Transmission</b>	
	Change of <b>Figure 13-92. Example of Contents of Registers for Data Transmission of Simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21)</b> and addition of <b>Note</b>	
	Change of <b>Figure 13-94. Flowchart of Data Transmission</b>	
	Change of <b>Figure 13-95. Example of Contents of Registers for Data Reception of Simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21)</b> and addition of <b>Note</b>	
	Change of <b>Figure 13-96. Timing Chart of Data Reception</b>	
	Change of <b>Figure 13-97. Flowchart of Data Reception</b> and addition of <b>Caution</b>	

Edition	Description	Chapter
2nd edition	Change of <b>Figure 13-99. Flowchart of Stop Condition Generation</b>	CHAPTER 13 SERIAL ARRAY UNIT
	Addition of setting and <b>Note</b> to <b>Table 13-4. Operating Clock Selection</b>	
	Addition of <b>13.9 Relationship Between Register Settings and Pins</b>	
	Change of <b>Figure 16-9. Example of Setting for UART Consecutive Reception + ACK Transmission</b>	CHAPTER 16 DMA CONTROLLER
	Additions of description to <b>16.6 (4) DMA pending instruction</b>	
	Change of <b>Table 17-1. Interrupt Source List</b>	CHAPTER 17 INTERRUPT FUNCTIONS
	Change of A/D converter and serial interface (IIC0) in <b>Table 19-1. Operating Statures in HALT Mode (2/2)</b>	CHAPTER 19 STANDBY FUNCTION
	Additions of <b>Note</b> to <b>Figure 19-3. HALT Mode Release by Reset by Interrupt Request Generation</b>	
	Change of <b>Figure 19-4. HALT Mode Release by Reset</b>	
	Additions of <b>Notes 2</b> to <b>Figure 19-5. Operation Timing When STOP Mode Is Released</b>	
	Addition of <b>Note</b> to <b>Figure 19-6. STOP Mode Release by Interrupt Request Generation</b>	
	Change of reset processing in <b>Figure 19-7. STOP Mode Release by Reset</b>	
	Change of description in <b>(4)</b>	
	Change of <b>Figure 20-2. Timing of Reset by RESET Input</b>	
	Change of <b>Figure 20-4. Timing of Reset in STOP Mode by RESET Input</b>	
	Change of <b>Caution 2</b> in <b>Figure 20-5. Format of Reset Control Flag Register (RESF)</b>	
	Change of <b>Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)</b>	CHAPTER 21 POWER-ON-CLEAR CIRCUIT
	Change of <b>Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)</b> and addition of <b>Notes 5</b>	
	Change of <b>Figure 21-3. Example of Software Processing After Reset Release</b>	
	Change of <b>Note 4</b> in <b>Figure 22-2. Format of Low-Voltage Detection Register (LVIM)</b> and addition of <b>Cautions 3</b>	CHAPTER 22 LOW-VOLTAGE DETECTOR
	Change of <b>Cautions 2</b> in <b>Figure 22-3. Format of Low-Voltage Detection Level Select Register (LVIS)</b>	
	Change of <5> in <b>22.4.1 (1) (a)</b>	
	Change of <b>Notes 2</b> in <b>Figure 22-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 1)</b>	
	Change of description and <b>Caution</b> in <b>22.4.1 (1) (b)</b>	
	Change of <b>Figure 22-6. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 0)</b> and <b>Note</b>	
	Change of <4> in <b>22.4.1 (2)</b>	
	Change of <b>Notes 2</b> in <b>Figure 22-7. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 1)</b>	
	Change of <5> in <b>22.4.2 (1)</b>	
	Additions of <b>Notes 3</b> to <b>Figure 22-8. Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 1)</b>	

Edition	Description	Chapter
2nd edition	Change of description and <b>Caution</b> in <b>22.4.2 (1) (b)</b>	CHAPTER 22 LOW-VOLTAGE DETECTOR
	Change of <b>Figure 22-9. Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 0)</b> and addition of <b>Note</b>	
	Change of <4> in <b>22.4.2 (2)</b>	
	Addition of <b>Notes 3</b> to <b>Figure 22-10. Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 1)</b>	
	Change of <b>Figure 22-11. Example of Software Processing After Reset Release</b>	
	Change of "MAX." to "MIN." of minimum pulse width	
	Change of <b>23.1 Regulator Overview</b>	CHAPTER 23 REGULATOR
	Addition of <b>Notes 3</b> to <b>Figure 23-1. Format of Regulator Mode Control Register (RMC)</b>	
	Change of description in <b>24.1.1 (2) 000C1H/010C1H</b>	CHAPTER 24 OPTION BYTE
	Change of <b>Figure 24-2. Format of User Option Byte (000C1H/010C1H)</b> and <b>Cautions 2</b>	
	Change of <b>25.4.1 FLMD0 pin</b>	CHAPTER 25 FLASH MEMORY
	Change of description in <b>25.4.5 REGC pin</b>	
	Addition of <b>Cautions 4</b> to <b>25.8 Flash Memory Programming by Self-Programming</b> and change of <b>Remark</b>	
	Change of <b>Figure 25-10. Flow of Self Programming (Rewriting Flash Memory)</b> , and addition of <b>Remark</b>	
	Addition of <b>26.3 Securing of user resources</b>	CHAPTER 26 ON-CHIP DEBUGGING
	Addition of <b>27.3 Securing of user resources</b>	CHAPTER 27 BCD CORRECTION CIRCUIT
	Addition of addr5 to <b>Table 28-2. Symbols in "Operation" Column</b>	CHAPTER 28 INSTRUCTION SET
	Change of operation of CALLT in <b>Table 28-5. Operation List (15/17)</b>	
	<b>Absolute Maximum Ratings</b> <ul style="list-style-type: none"> <li>• Change of conditions in Supply voltage (EVDD0, EVDD1)</li> <li>• Change of conditions in Supply voltage (EVSS0, EVSS1)</li> <li>• Change of Input voltage</li> <li>• Change of Output voltage</li> <li>• Change of rating of Analog output voltage (VAO)</li> <li>• Change of conditions in Output current, high (IOH1)</li> <li>• Change of conditions in Output current, low (IOL1)</li> </ul>	CHAPTER 29 ELECTRICAL SPECIFICATIONS (TARGET)
	Change of <b>Notes 1</b> in <b>Internal Oscillator Characteristics</b>	

Edition	Description	Chapter
2nd edition	<p><b>DC Characteristics</b></p> <ul style="list-style-type: none"> <li>• Addition of duty and <b>Notes 2</b> and change of conditions in Output current, high (I<sub>OH1</sub>)</li> <li>• Change of conditions in Output current, high (I<sub>OH2</sub>)</li> <li>• Addition of duty and <b>Notes 2</b> and change of conditions and MAX. value in Output current, high (I<sub>OL1</sub>)</li> <li>• Change of conditions in Output current, low (I<sub>OL2</sub>)</li> <li>• Change of Input voltage, high (V<sub>IH3</sub>)</li> <li>• Change of conditions in Input voltage, high (V<sub>IH4</sub>)</li> <li>• Change of conditions in Input voltage, high (V<sub>IH5</sub>)</li> <li>• Change of <b>Cautions 2</b></li> <li>• Change of Input voltage, low (V<sub>IL3</sub>)</li> <li>• Change of conditions of Input voltage, low (V<sub>IL4</sub>)</li> <li>• Change of conditions of Input voltage, low (V<sub>IL5</sub>)</li> <li>• Change of conditions in Output voltage, high (V<sub>OH1</sub>)</li> <li>• Change of Output voltage, high (V<sub>OH2</sub>)</li> <li>• Change of conditions in Output voltage, low (V<sub>OL1</sub>)</li> <li>• Change of conditions in Output voltage, low (V<sub>OL2</sub>)</li> <li>• Change of conditions of Input leakage current, high (I<sub>LIH2</sub>)</li> <li>• Change of conditions of Input leakage current, high (I<sub>LIH3</sub>)</li> <li>• Change of conditions of Input leakage current, low (I<sub>LIL2</sub>)</li> <li>• Change of conditions of Input leakage current, low (I<sub>LIL3</sub>)</li> <li>• Change of Pull-up resistance value and Pull-down resistance value</li> <li>• Addition of FLMD0 pin external pull-down resistance (R<sub>FLMD0</sub>) and <b>Note</b></li> <li>• Addition of Supply current and operating current of macros</li> </ul> <hr/> <p><b>AC Characteristics</b></p> <p><b>(1) Basic operation</b></p> <ul style="list-style-type: none"> <li>• Change of Instruction cycle (minimum instruction execution time) (T<sub>CV</sub>)</li> <li>• Change of MAX. value in External main system clock input high-level width, low-level width (t<sub>EXH</sub>, t<sub>EXL</sub>)</li> <li>• Change of MIN. value in T100 to T107 and T110 to T113 input high-level width, low-level width (t<sub>TIH</sub>, t<sub>TIL</sub>)</li> <li>• Change of <b>Remarks 1, 2</b></li> <li>• Addition of figures of <b>Minimum instruction execution time during main system clock operation</b> and <b>Minimum instruction execution time during self programming mode</b> in <b>(1) Basic operation</b></li> <li>• Change of title in <b>AC Timing Test Points</b></li> </ul> <p>Addition of <b>(2) External bus interface</b></p> <p>Addition of <b>(3) Serial interface: Serial array unit</b></p> <p>Addition of <b>(4) Serial interface: IIC0</b></p> <p>Addition of <b>(5) Serial interface: On-chip debug (UART)</b></p> <hr/> <p>Modification of condition in upper part of table of <b>A/D Converter Characteristics</b></p> <hr/> <p>Change of <b>A/D Converter Characteristics</b></p> <hr/> <p>Addition of Detection delay time to <b>POC Circuit Characteristics</b></p> <hr/> <p>Change of <b>Supply Voltage Rise Time</b></p>	CHAPTER 29 ELECTRICAL SPECIFICATIONS (TARGET)



Edition	Description	Chapter
2nd edition	Change of figures and figure title in <b>Supply Voltage Rise Time Timing</b>	CHAPTER 29 ELECTRICAL SPECIFICATIONS (TARGET)
	Addition of Detection delay time to <b>LVI Circuit Characteristics</b>	
	Change of <b>Flash Memory Programming Characteristics</b>	
	Change of <b>A.4.1 When using flash memory programmer FG-FP4 and FL-PR4</b>	APPENDIX A DEVELOPMENT TOOLS
	Change of <b>A.4.2 When using on-chip debug emulator with programming function QB-MINI2</b>	
	Change of <b>A.5.2 When using on-chip debug emulator with programming function QB-MINI2</b>	
4th edition	Addition of expanded-specification products, $\mu$ PD78F1174A, 78F1175A, 78F1176A, 78F1177A, 78F1178A Addition of (A) grade products of expanded-specification products, $\mu$ PD78F1174A(A), 78F1175A(A), 78F1176A(A), 78F1177A(A), 78F1178A(A)	Throughout
	Change of related documents	INTRODUCTION
	Addition of <b>1.1 Differences Between Conventional-Specification Products (<math>\mu</math>PD78F117x) and Expanded-Specification Products (<math>\mu</math>PD78F117xA)</b>	CHAPTER 1 OUTLINE
	Addition of <b>Caution 4 to 1.5 Pin Configuration (Top View)</b>	
	Modification of <b>1.7 Block Diagram</b>	
	Change of description in <b>2.2.17 AV<sub>REF0</sub></b>	CHAPTER 2 PIN FUNCTIONS
	Change of description in <b>2.2.18 AV<sub>REF1</sub></b>	
	Change of description in <b>2.2.20 RESET</b>	
	Change of I/O circuit type of P97 and P127 and descriptions of AV <sub>REF0</sub> , AV <sub>REF1</sub> , and RESET pin in <b>Table 2-4 Connection of Unused Pins</b>	
	Addition of <b>Note to Figures 3-1 to 3-5</b>	CHAPTER 3 CPU ARCHITECTURE
	Change of figure in <b>Remark of 3.1 Memory Space</b>	
	Change of description in <b>3.1.1 (1) Vector table area</b>	
	Change of description in <b>3.1.2 Mirror area</b>	
	Change of description and addition and change of <b>Caution in 3.1.3 Internal data memory space</b>	
	Addition of <b>Cautions to 3.2.1 (3) Stack pointer (SP)</b>	
	Modification of <b>Table 3-5 SFR List</b>	CHAPTER 4 PORT FUNCTIONS
	Change of <b>Figure 4-26 Block Diagram of P52 to P57</b>	
	Addition of description of PU6	
	Modification of <b>Figure 4-31 Block Diagram of P80 to P87</b>	
	Addition of <b>Caution 3 to Figure 4-55 Format of A/D Port Configuration Register (ADPC)</b>	
	Change of <b>Notes of Table 4-6 Settings of Port Mode Register and Output Latch When Using Alternate Function</b>	CHAPTER 5 EXTERNAL BUS INTERFACE
	Change of <b>5.5 Number of Instruction Execution Clocks and Instruction Wait Clocks for Fetch Access</b>	
	Change of description in <b>5.6 (4) WAIT pin (alternate function: P06)</b>	
	Addition of <b>5.6 (6) CLKOUT pin (alternate function: P05)</b>	

Edition	Description	Chapter
4th edition	Change of <b>Cautions 3 and 5 in Figure 6-8 Format of Operation Speed Mode Control Register (OSMC)</b>	CHAPTER 6 CLOCK GENERATOR
	Change of <b>Figure 6-13 Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1))</b>	
	Change of <b>Figure 6-14 Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0))</b> and description of <1>	
	Change of <b>6.6.3 (1) &lt;1&gt; Setting P123/XT1 and P124/XT2 pins (CMC register)</b>	
	Addition of description to <b>6.6.3 (3) &lt;1&gt; Confirming the CPU clock status (CKC register)</b>	
	Change of and deletion of <b>Note in Figure 6-15 CPU Clock Status Transition Diagram</b>	
	Addition of <b>Note to Table 6-5. Changing CPU Clock</b>	
	Change of <b>Table 6-6 Maximum Time Required for Main System Clock Switchover</b>	
	4th edition	
Change of <b>Caution in Figure 7-5. Format of Peripheral Enable Register 0 (PER0)</b>		
Change of description of CCSmn bit in <b>Figure 7-7 Format of Timer Mode Register mn (TMRmn)</b>		
Change of <b>Figure 7-21 Format of Timer Output Mode Register m (TOMm)</b>		
Change of description in <b>7.4.3 (1) Changing values set in registers TOM, TOEm, TOLm, and TOMm during timer operation</b>		
Addition of description to <b>7.7.1 (1) Interval timer</b>		
Change of <b>Figure 7-37 Block Diagram of Operation as Interval Timer/Square Wave Output</b>		
Addition of <b>(2) When fsub/4 is selected as count clock to Figure 7-39 Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output</b>		
Change of <b>Figure 7-40 Operation Procedure of Interval Timer/Square Wave Output Function</b>		
Change of description during operation in <b>Figure 7-44 Operation Procedure When External Event Counter Function Is Used</b>		
Change of channel number in <b>7.7.3 Operation as frequency divider</b>		
Change of description during operation in <b>Figure 7-48 Operation Procedure When Frequency Divider Function Is Used</b>		
Change of description during operation in <b>Figure 7-52 Operation Procedure When Input Pulse Interval Measurement Function Is Used</b>		
Change of description during operation in <b>Figure 7-56 Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used</b>		
Change of description during operation in <b>Figure 7-61 Operation Procedure When PWM Function Is Used</b>		
Change of description during operation in <b>Figure 7-66 Operation Procedure of One-Shot Pulse Output Function</b>		
Change of description during operation in <b>Figure 7-71 Operation Procedure When Multiple PWM Output Function Is Used</b>		

Edition	Description	Chapter
4th edition	Change of <b>Note</b> in <b>Figure 8-2 Format of Peripheral Enable Register 0 (PER0)</b>	CHAPTER 8 REAL-TIME COUNTER
	Change of <b>Figure 8-3 Format of Real-Time Counter Control Register 0 (RTCC0)</b>	
	Change of description and <b>Caution</b> in <b>Figure 8-4 Format of Real-Time Counter Control Register 1 (RTCC1)</b>	
	Addition of <b>Caution 3</b> to <b>Figure 8-5 Format of Real-Time Counter Control Register 2 (RTCC2)</b>	
	Change of description in <b>8.3 (7) Minute count register (MIN), (8) Hour count register (HOUR), (9) Day count register (DAY), (11) Month count register (MONTH), and (12) Year count register (YEAR)</b>	
	Change of <b>Table 8-2 Displayed Time Digits</b>	
	Addition of <b>Caution</b> to <b>Figure 8-11 Format of Week Count Register (WEEK)</b>	
	Change of description in <b>8.3 (13) Watch error correction register (SUBCUD)</b>	
	Deletion of <b>Caution</b> in <b>8.3 (16) Alarm week register (ALARMWW)</b>	
	Addition of <b>Note 2</b> to <b>Figure 8-18 Procedure for Starting Operation of Real-Time Counter</b>	
	Addition of <b>8.4.2 Shifting to STOP mode after starting operation</b>	
	Addition of <b>8.4.5 1 Hz output of real-time counter</b>	
	Addition of <b>8.4.6 32.768 kHz output of real-time counter</b>	
	Addition of <b>8.4.7 512 Hz, 16.384 kHz output of real-time counter</b>	
Addition of <b>8.4.8 Example of watch error correction of real-time counter</b>		
	Change of <b>Caution 3</b> in <b>Table 9-4 Setting Window Open Period of Watchdog Timer</b>	CHAPTER 9 WATCHDOG TIMER
	Change of description in <b>11.2 (9) AV<sub>REF0</sub> pin</b>	CHAPTER 11 A/D CONVERTER
	Change of <b>Table 11-3 A/D Conversion Time Selection</b>	
	Change of <b>Figure 11-6. A/D Converter Sampling and A/D Conversion Timing</b>	
	Addition of <b>Caution 3</b> to <b>Figure 11-10 Format of A/D Port Configuration Register (ADPC)</b>	
	Addition of <b>11.5 Temperature Sensor Function</b>	
	Addition of <b>11.7 (2) Reducing current when A/D converter is stopped</b>	
	Addition of <b>12.2 (1) AV<sub>REF1</sub> pin</b>	CHAPTER 12 D/A CONVERTER
	Addition of <b>12.3 (4) Port mode register 11 (PM11)</b>	
	Change of description in <b>12.4.1 Operation in normal mode</b>	
	Change of description in <b>12.4.2 Operation in real-time output mode</b>	
	Addition of <b>Note</b> to <b>13.1.3 Simplified I<sup>2</sup>C (IIC10, IIC11, IIC20 IIC21)</b>	CHAPTER 13 SERIAL ARRAY UNIT
	Change of <b>Note 2</b> in <b>Figure 13-5 Format of Serial Clock Select Register m (SPSm)</b>	
	Change of <b>Figure 13-7 Format of Serial Communication Operation Setting Register mn (SCRmn)</b>	
	Change of <b>Figure 13-26 Procedure for Stopping Master Transmission</b>	
	Change of <b>Figure 13-28 Timing Chart of Master Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)</b>	

Edition	Description	Chapter
4th edition	Change of <b>Figure 13-30 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)</b>	CHAPTER 13 SERIAL ARRAY UNIT (continuation)
	Modification of <b>Figure 13-36 Timing Chart of Master Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)</b>	
	Change of <b>Figure 13-40 Procedure for Stopping Master Transmission/Reception</b>	
	Change of <b>Figure 13-41 Procedure for Resuming Master Transmission/Reception</b>	
	Modification of <b>Figure 13-42 Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)</b>	
	Modification of <b>Figure 13-44 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)</b>	
	Change of transfer rate in <b>13.5.4 Slave transmission</b>	
	Change of <b>Figure 13-48 Procedure for Stopping Slave Transmission</b>	
	Change of <b>Figure 13-49 Procedure for Resuming Slave Transmission</b>	
	Change of <b>Figure 13-50 Timing Chart of Slave Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)</b>	
	Change of <b>Figure 13-52 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)</b>	
	Change of <b>Figure 13-53 Flowchart of Slave Transmission (in Continuous Transmission Mode)</b>	
	Change of transfer rate in <b>13.5.5 Slave reception</b>	
	Change of <b>(b) Serial output enable register m (SOEm) in Figure 13-54. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21)</b>	
	Modification of <b>Figure 13-58 Timing Chart of Slave Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)</b>	
	Change of transfer rate in <b>13.5.6 Slave transmission/reception</b>	
	Change of <b>Figure 13-62 Procedure for Stopping Slave Transmission/Reception</b>	
	Change of <b>Figure 13-63 Procedure for Resuming Slave Transmission/Reception</b>	
	Modification of <b>Figure 13-64 Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)</b>	
	Modification of <b>Figure 13-66 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)</b>	
	Change of <b>13.5.7 Calculating transfer clock frequency</b>	
	Change of <b>Note 2 in Table 13-2 Selection of Operation Clock</b>	
	Addition of <b>Caution to 13.6 Operation of UART (UART0, UART1, UART2, UART3) Communication</b>	
	Change of <b>Figure 13-70 Procedure for Stopping UART Transmission</b>	
	Change of <b>Figure 13-72 Timing Chart of UART Transmission (in Single-Transmission Mode)</b>	
	Change of <b>Figure 13-74 Timing Chart of UART Transmission (in Continuous Transmission Mode)</b>	
	Change of <b>13.6.2 UART reception</b>	

Edition	Description	Chapter
4th edition	Change of <b>(b) Serial output enable register m (SOEm) in Figure 13-76 Example of Contents of Registers for UART Reception of UART (UART0, UART1, UART2, UART3)</b>	CHAPTER 13 SERIAL ARRAY UNIT (continuation)
	Modification of <b>Figure 13-80 Timing Chart of UART Reception</b>	
	Modification of transfer data length in <b>13.6.3 LIN transmission</b>	
	Change of <b>Note 2 in Figure 13-82 Transmission Operation of LIN</b>	
	Change of <b>Note 2 in Table 13-3 Selection of Operation Clock</b>	
	Addition of <b>Note to 13.7 Operation of Simplified I<sup>2</sup>C (IIC10, IIC11, IIC20, IIC21) Communication</b>	
	Addition of <b>Note to 13.7.1 Address field transmission</b>	
	Change of <b>Figure 13-89 Initial Setting Procedure for Address Field Transmission</b>	
	Change of <b>Figure 13-90 Timing Chart of Address Field Transmission</b>	
	Addition of <b>Note to 13.7.2 Data transmission</b>	
	Change of <b>Figure 13-93 Timing Chart of Data Transmission</b>	
	Addition of <b>Note to 13.7.3 Data reception</b>	
	Change of <b>Figure 13-96 Timing Chart of Data Reception</b>	
	Change of <b>Figure 13-97 Flowchart of Data Reception and change of Caution</b>	
	Change of <b>Figure 13-98 Timing Chart of Stop Condition Generation</b>	
	Change of <b>Note 2 in Table 13-4 Selection of Operation Clock</b>	
	Change of <b>Note in Figure 14-6 Format of IIC Control Register 0 (IICC0)</b>	CHAPTER 14 SERIAL INTERFA CE IIC0
	Change of <b>Table 14-2 Selection Clock Setting</b>	
	Change of <b>Table 14-3 Selection Clock Setting</b>	
	Change of <b>Table 14-5 Extension Code Bit Definitions</b>	
	Change of <b>Figure 14-24 Master Operation in Single-Master System</b>	
	Change of <b>Figure 14-25 Master Operation in Multi-Master System</b>	
	Change of <b>Figure 14-26 Slave Operation Flowchart</b>	
	Change of <b>Figures 14-28 and 14-29</b>	
	Change of <b>Figure 16-5 Format of DMA Operation Control Register n (DRCn)</b>	CHAPTER 16 DMA CONTROLLER
	Addition of <b>Note to Table 16-2 Response Time of DMA Transfer</b>	
	Change of description in <b>17.2 Interrupt Sources and Configuration</b>	CHAPTER 17 INTERRUPT FUNCTIONS
	Change of <b>Table 17-1 Interrupt Source List</b>	
	Change of <b>Caution 2 in 18.3 (1) Key return mode register (KRM)</b>	CHAPTER 18 KEY INTERRUPT FUNCTION

Edition	Description	Chapter
4th edition	Change of external bus interface in <b>Table 19-1 Operating Statuses in HALT Mode</b>	CHAPTER 19 STANDBY FUNCTION
	Change of <b>Note</b> in <b>Figure 19-3 HALT Mode Release by Interrupt Request Generation</b>	
	Change of external bus interface in <b>Table 19-2 Operating Statuses in STOP Mode</b>	
	Change of <b>Figure 19-5 Operation Timing When STOP Mode Is Released (Release by Unmasked Interrupt Request)</b>	
	Addition of <b>Note</b> to <b>Figure 19-6 STOP Mode Release by Interrupt Request Generation</b>	
	Deletion of <b>Note</b> in <b>21.1 Functions of Power-on-Clear Circuit</b>	CHAPTER 21 POWER-ON-CLEAR CIRCUIT
	Deletion of <b>Note</b> in <b>21.3 Operation of Power-on-Clear Circuit</b>	
	Deletion of <b>Note 6</b> in <b>(1) When LVI is OFF upon power application (option byte: LVIOFF = 1) in Figure 21-2 Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector</b>	
	Deletion of <b>Note 3</b> in <b>(2) When LVI is ON upon power application (option byte: LVIOFF = 0) in Figure 21-2 Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector</b>	
	Deletion of <b>Note</b> in <b>22.1 Functions of Low-Voltage Detector</b>	CHAPTER 22 LOW-VOLTAGE DETECTOR
	Deletion of <b>Note 2</b> in <b>Figure 22-3 Format of Low-Voltage Detection Level Select Register (LVIS)</b>	
	Deletion of <b>Note</b> in <b>22.4 Operation of Low-Voltage Detector</b>	
	Addition of description to <b>24.4 Setting of Option Byte</b>	CHAPTER 24 OPTION BYTE
	Addition of PG-FP5, FL-PR5, and QB-MINI2 as dedicated flash memory programmers	CHAPTER 25 FLASH MEMORY
	Change of <b>Figure 25-6 Format of Background Event Control Register (BECTL)</b>	
	Change of <b>Table 25-4 Communication Modes</b>	
	Addition of <b>25.8 Processing Time of Each Command When Using PG-FP4 or PG-FP5 (Reference Values)</b>	
	Addition of <b>Caution 5</b> to <b>25.9 Flash Memory Programming by Self-Programming</b>	
	Change of description in <b>25.9.2 Flash shield window function</b>	
	Change of <b>Caution</b> in <b>26.1 Connecting QB-MINI2 to 78K0R/KH3</b>	CHAPTER 26 ON-CHIP DEBUG FUNCTION
	Addition of <b>Caution</b> to <b>Figure 26-1 Connection Example of QB-MINI2 and 78K0R/KH3</b>	
	Change of <b>Table 26-1 Differences Between 1-Line Mode and 2-Line Mode</b>	
	Change of <b>Table 28-1 Operand Identifiers and Specification Methods</b> and change of <b>Remark</b>	CHAPTER 28 INSTRUCTION SET
	Change of description in <b>28.1.4 PREFIX instruction</b>	
	Change of <b>Remark 2</b> in <b>Table 28-5 Operation List</b>	
	Change of "Clocks" column of BTCLR instruction in <b>Table 28-5 Operation List</b>	

Edition	Description	Chapter
4th edition	Change of specifications of $\mu$ PD78F1174A, 78F1175A, 78F1176A, 78F1177A, and 78F1178A from target specifications to formal specifications	CHAPTER 29 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS)
	Addition of recommended oscillator constants	
	Change of "Conditions" column and MAX. values of output current, low ( $I_{OL1}$ ) in <b>DC Characteristics</b>	
	Change of "Conditions" column of input voltage, high ( $V_{IH1}$ ) in <b>DC Characteristics</b>	
	Change of "Conditions" column of input voltage, low ( $V_{IL1}$ ) in <b>DC Characteristics</b>	
	Change of "Conditions" column of output voltage, low ( $V_{OL1}$ ) in <b>DC Characteristics</b>	
	Change of supply current value in <b>DC Characteristics</b>	
	Modification of "Conditions" column of instruction cycle and change of external main system clock frequency and external main system clock input high-level width, low-level width in <b>(1) Basic operation in AC Characteristics</b>	
	Addition of $t_{DAR1}$ and expanded-specification product specifications to <b>(a) Read/write cycle (CLKOUT synchronous) in (2) External bus interface</b>	
	Addition of $t_{DAR1}$ to timing diagram in <b>Read/write cycle (CLKOUT synchronous): In multiplexed bus mode in (2) External bus interface</b>	
	Addition of <b>(b) Read/write cycle (CLKOUT asynchronous) to (2) External bus interface</b>	
	Addition of expanded-specification product specifications to <b>(d) During communication at same potential (simplified I<sup>2</sup>C mode) in (3) Serial interface: Serial array unit</b>	
	Change of overall error and integral linearity error values in <b>A/D Converter Characteristics</b>	
	Addition of A/D converter characteristics of expanded-specification products	
	Addition of <b>Temperature sensor</b>	
Addition of <b>Remark</b> to <b>D/A Converter Characteristics</b>		
Change of $V_{DD}$ supply current value and number of rewrites and addition of expanded-specification product characteristics in <b>Flash Memory Programming Characteristics</b>		
	Addition of chapter	CHAPTER 30 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS) (TARGET)
	Addition of chapter	CHAPTER 32 RECOMMENDED SOLDERING CONDITIONS
	Change of <b>A.5.2 When using on-chip debug emulator with programming function QB-MINI2</b>	APPENDIX A DEVELOPMENT TOOLS
	Addition of chapter	APPENDIX B LIST OF CAUTIONS

*For further information,  
please contact:*

**NEC Electronics Corporation**

1753, Shimonumabe, Nakahara-ku,  
Kawasaki, Kanagawa 211-8668,  
Japan  
Tel: 044-435-5111  
<http://www.necel.com/>

**[America]**

**NEC Electronics America, Inc.**

2880 Scott Blvd.  
Santa Clara, CA 95050-2554, U.S.A.  
Tel: 408-588-6000  
800-366-9782  
<http://www.am.necel.com/>

**[Europe]**

**NEC Electronics (Europe) GmbH**

Arcadiastrasse 10  
40472 Düsseldorf, Germany  
Tel: 0211-65030  
<http://www.eu.necel.com/>

**Hanover Office**

Podbielskistrasse 166 B  
30177 Hannover  
Tel: 0 511 33 40 2-0

**Munich Office**

Werner-Eckert-Strasse 9  
81829 München  
Tel: 0 89 92 10 03-0

**Stuttgart Office**

Industriestrasse 3  
70565 Stuttgart  
Tel: 0 711 99 01 0-0

**United Kingdom Branch**

Cygnus House, Sunrise Parkway  
Linford Wood, Milton Keynes  
MK14 6NP, U.K.  
Tel: 01908-691-133

**Succursale Française**

9, rue Paul Dautier, B.P. 52  
78142 Velizy-Villacoublay Cédex  
France  
Tel: 01-3067-5800

**Sucursal en España**

Juan Esplandiú, 15  
28007 Madrid, Spain  
Tel: 091-504-2787

**Tyskland Filial**

Täby Centrum  
Entrance S (7th floor)  
18322 Täby, Sweden  
Tel: 08 638 72 00

**Filiale Italiana**

Via Fabio Filzi, 25/A  
20124 Milano, Italy  
Tel: 02-667541

**Branch The Netherlands**

Steijgerweg 6  
5616 HS Eindhoven  
The Netherlands  
Tel: 040 265 40 10

**[Asia & Oceania]**

**NEC Electronics (China) Co., Ltd**

7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian  
District, Beijing 100083, P.R.China  
Tel: 010-8235-1155  
<http://www.cn.necel.com/>

**Shanghai Branch**

Room 2509-2510, Bank of China Tower,  
200 Yincheng Road Central,  
Pudong New Area, Shanghai, P.R.China P.C:200120  
Tel:021-5888-5400  
<http://www.cn.necel.com/>

**Shenzhen Branch**

Unit 01, 39/F, Excellence Times Square Building,  
No. 4068 Yi Tian Road, Futian District, Shenzhen,  
P.R.China P.C:518048  
Tel:0755-8282-9800  
<http://www.cn.necel.com/>

**NEC Electronics Hong Kong Ltd.**

Unit 1601-1613, 16/F., Tower 2, Grand Century Place,  
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: 2886-9318  
<http://www.hk.necel.com/>

**NEC Electronics Taiwan Ltd.**

7F, No. 363 Fu Shing North Road  
Taipei, Taiwan, R. O. C.  
Tel: 02-8175-9600  
<http://www.tw.necel.com/>

**NEC Electronics Singapore Pte. Ltd.**

238A Thomson Road,  
#12-08 Novena Square,  
Singapore 307684  
Tel: 6253-8311  
<http://www.sg.necel.com/>

**NEC Electronics Korea Ltd.**

11F., Samik Lavied'or Bldg., 720-2,  
Yeoksam-Dong, Kangnam-Ku,  
Seoul, 135-080, Korea  
Tel: 02-558-3737  
<http://www.kr.necel.com/>