User's Manual

16

78K0R/Kx3-C

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.2.00 Sep 2011

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.

NOTES FOR CMOS DEVICES

(1) VOLTAGE APPLICATION WAVEFORM AT INPUT PIN: Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (MAX) and VIH (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (MAX) and VIH (MIN).

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

(3) PRECAUTION AGAINST ESD: A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

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

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

(6) INPUT OF SIGNAL DURING POWER OFF STATE : Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the 78K0R/Kx3-C and design and develop application systems and programs for these devices.
The target products are as follows.

- 78K0R/KF3-C: μPD78F1846A, 78F1847A
- 78K0R/KG3-C: μPD78F1848A, 78F1849A

Purpose

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

Organization

The 78K0R/Kx3-C manual is separated into two parts: this manual and the instructions edition (common to the 78K0R Microcontroller).

78K0R/Kx3-C User's Manual (This Manual)	**78K0R Microcontroller User's Manual Instructions**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

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

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what." field.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R.
- To know details of the 78K0R Microcontroller 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:	$\overline{\times\times\times}$ (overscore over pin and signal name)	
Note:	Footnote for item marked with **Note** in the text	
Caution:	Information requiring particular attention	
Remark:	Supplementary information	
Numerical representations:	Binary	···×××× or ××××B
	Decimal	···××××
	Hexadecimal	···××××H

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

Documents Related to Devices

Document Name	Document No.
78K0R/Kx3-C User's Manual	This manual
78K0R Microcontroller Instructions User's Manual	R01US0029E
78K0R Microcontroller Self Programming Library Type02 User's Manual[Note]	U19193E

Note This document is classified under engineering management. Contact a Renesas 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	U18010E
PM+ Ver. 6.30		U18416E
ID78K0R-QB Ver. 3.20 Integrated Debugger	Operation	U17839E
CubeSuite+ V1.00.00 Integrated Development Environment[Note]	Start	R20UT0545E
	78K0R Design	R20UT0547E
	RL78, 78K0R Coding	R20UT0552E
	RL78, 78K0R Build	R20UT0556E
	78K0R Debug	R20UT0560E
	Analysis	R20UT0563E
	Message	R20UT0407E

Note Confirm the latest information of CubeSuite+ by referring to the following website.
http://www.renesas.com/cubesuite+

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

Document Name	Document No.
QB-MINI2 On-Chip Debug Emulator with Programming Function	R02UT0449E
QB-78K0RIX3C In-Circuit Emulator	U19324E

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

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP5 Flash Memory Programmer User's Manual	R02UT0008E

Other Documents

Document Name	Document No.
RENESAS MICROCOMPUTER GENERAL CATALOG	R01CS0001E
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

Note See the "Semiconductor Device Mount Manual" website (http://www.renesas.com/prod/package/manual/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.

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

[MEMO]

CONTENTS

CHAPTER 1 OUTLINE 19

1.1 Features 19
1.2 Applications 20
1.3 Ordering Information 20
1.4 Pin Configuration (Top View) 21
1.4.1 78K0R/KF3-C 21
1.4.2 78K0R/KG3-C 23
1.5 78K0R/Kx3-C Microcontroller Lineup 25
1.6 Block Diagram 26
1.6.1 78K0R/KF3-C 26
1.6.2 78K0R/KG3-C 27
1.7 Outline of Functions 28

CHAPTER 2 PIN FUNCTIONS 30

2.1 Pin Function List 30
2.1.1 78K0R/KF3-C 31
2.1.2 78K0R/KG3-C 36
2.2 Description of Pin Functions 41
2.2.1 P00 to P06 (port 0) 41
2.2.2 P10 to P17 (port 1) 42
2.2.3 P20 to P27 (port 2) 43
2.2.4 P30, P31 (port 3) 44
2.2.5 P40 to P47 (port 4) 44
2.2.6 P50 to P57 (port 5) 46
2.2.7 P60 to P67 (port 6) 47
2.2.8 P70 to P77 (port 7) 48
2.2.9 P80 to P87 (port 8) (78K0R/KG3-C only) 48
2.2.10 P90, P91 (port 9) 48
2.2.11 P110, P111 (port 11) 49
2.2.12 P120 to P124 (port 12) 49
2.2.13 P130, P131 (port 13) 50
2.2.14 P140 to P145 (port 14) 51
2.2.15 P150 to P157 (port 15) 52
2.2.16 AV_{REF} 53
2.2.17 AV_{SS} 53
2.2.18 $\overline{RESET}$ 53
2.2.19 REGC 53
2.2.20 V_{DD}, EV_{DD0}, EV_{DD1} 54

2.2.21 V_{SS}, EV_{SS0}, EV_{SS1} 54
2.2.22 FLMD0 54
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins 55
2.3.1 78K0R/KF3-C 55
2.3.2 78K0R/KG3-C 58

CHAPTER 3 CPU ARCHITECTURE 63

3.1 Memory Space 63
3.1.1 Internal program memory space 67
3.1.2 Mirror area 69
3.1.3 Internal data memory space 70
3.1.4 Special function register (SFR) area 71
3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area 71
3.1.6 Data memory addressing 72
3.2 Processor Registers 74
3.2.1 Control registers 74
3.2.2 General-purpose registers 76
3.2.3 ES and CS registers 78
3.2.4 Special function registers (SFRs) 79
3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers) 85
3.3 Instruction Address Addressing 93
3.3.1 Relative addressing 93
3.3.2 Immediate addressing 93
3.3.3 Table indirect addressing 94
3.3.4 Register direct addressing 95
3.4 Addressing for Processing Data Addresses 96
3.4.1 Implied addressing 96
3.4.2 Register addressing 96
3.4.3 Direct addressing 97
3.4.4 Short direct addressing 98
3.4.5 SFR addressing 99
3.4.6 Register indirect addressing 100
3.4.7 Based addressing 101
3.4.8 Based indexed addressing 104
3.4.9 Stack addressing 105

CHAPTER 4 PORT FUNCTIONS 106

4.1 Port Functions 106
4.2 Port Configuration 111
4.2.1 Port 0 112
4.2.2 Port 1 118
4.2.3 Port 2 126
4.2.4 Port 3 128

4.2.5 Port 4 ... 129
4.2.6 Port 5 ... 136
4.2.7 Port 6 ... 140
4.2.8 Port 7 ... 145
4.2.9 Port 8 (78K0R/KG3-C only) ... 146
4.2.10 Port 9 ... 147
4.2.11 Port 11 ... 149
4.2.12 Port 12 ... 151
4.2.13 Port 13 ... 154
4.2.14 Port 14 ... 156
4.2.15 Port 15 ... 161
4.3 Registers Controlling Port Function ... 163
4.4 Port Function Operations ... 176
4.4.1 Writing to I/O port ... 176
4.4.2 Reading from I/O port ... 176
4.4.3 Operations on I/O port ... 176
4.4.4 Connecting to external device with different potential (2.5 V, 3 V) ... 177
4.5 Settings of PF11, PF6, Port Mode Register, and Output Latch When Using Alternate Function ... 179
4.6 Cautions on 1-Bit Manipulation Instruction for Port Register n (Pn) ... 183

CHAPTER 5 CLOCK GENERATOR ... 184

5.1 Functions of Clock Generator ... 184
5.2 Configuration of Clock Generator ... 185
5.3 Registers Controlling Clock Generator ... 187
5.4 System Clock Oscillator ... 202
5.4.1 X1 oscillator ... 202
5.4.2 XT1 oscillator ... 202
5.4.3 Internal high-speed oscillator ... 206
5.4.4 Internal low-speed oscillator ... 206
5.4.5 Prescaler ... 206
5.5 Clock Generator Operation ... 207
5.6 Controlling Clock ... 214
5.6.1 Example of controlling high-speed system clock ... 214
5.6.2 Example of controlling internal high-speed oscillation clock ... 217
5.6.3 Example of controlling subsystem clock ... 219
5.6.4 Example of controlling internal low-speed oscillation clock ... 221
5.6.5 CPU clock status transition diagram ... 222
5.6.6 Condition before changing CPU clock and processing after changing CPU clock ... 229
5.6.7 Time required for switchover of CPU clock and main system clock ... 231
5.6.8 Conditions before clock oscillation is stopped ... 232

CHAPTER 6 TIMER ARRAY UNIT 233

6.1 Functions of Timer Array Unit 233

6.1.1 Functions of each channel when it operates independently 233

6.1.2 Functions of each channel when it operates with another channel 234

6.2 Configuration of Timer Array Unit 235

6.3 Registers Controlling Timer Array Unit 241

6.4 Channel Output (TOmn pin) Control 266

6.4.1 TOmn pin output circuit configuration 266

6.4.2 TOmn Pin Output Setting 267

6.4.3 Cautions on Channel Output Operation 268

6.4.4 Collective manipulation of TOmn bits 271

6.4.5 Timer Interrupt and TOmn Pin Output at Operation Start 272

6.5 Channel Input (TImn Pin) Control 273

6.5.1 TImn edge detection circuit 273

6.6 Basic Function of Timer Array Unit 274

6.6.1 Overview of single-operation function and combination-operation function 274

6.6.2 Basic rules of combination-operation function 274

6.6.3 Applicable range of basic rules of combination-operation function 275

6.7 Operation of Timer Array Unit as Independent Channel 276

6.7.1 Operation as interval timer/square wave output 276

6.7.2 Operation as external event counter 283

6.7.3 Operation as frequency divider (channel 0 of unit 0 only) 286

6.7.4 Operation as input pulse interval measurement 290

6.7.5 Operation as input signal high-/low-level width measurement 294

6.8 Operation of Plural Channels of Timer Array Unit 298

6.8.1 Operation as PWM function 298

6.8.2 Operation as one-shot pulse output function 305

6.8.3 Operation as multiple PWM output function 312

CHAPTER 7 REAL-TIME COUNTER 320

7.1 Functions of Real-Time Counter 320

7.2 Configuration of Real-Time Counter 320

7.3 Registers Controlling Real-Time Counter 322

7.4 Real-Time Counter Operation 337

7.4.1 Starting operation of real-time counter 337

7.4.2 Shifting to STOP mode after starting operation 338

7.4.3 Reading/writing real-time counter 339

7.4.4 Setting alarm of real-time counter 341

7.4.5 1 Hz output of real-time counter 342

7.4.6 32.768 kHz output of real-time counter 342

7.4.7 512 Hz, 16.384 kHz output of real-time counter 342

7.4.8 Example of watch error correction of real-time counter 343

CHAPTER 8 WATCHDOG TIMER 348

8.1 Functions of Watchdog Timer 348
8.2 Configuration of Watchdog Timer 349
8.3 Register Controlling Watchdog Timer 350
8.4 Operation of Watchdog Timer 351
8.4.1 Controlling operation of watchdog timer 351
8.4.2 Setting overflow time of watchdog timer 352
8.4.3 Setting window open period of watchdog timer 353
8.4.4 Setting watchdog timer interval interrupt 354

CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER 355

9.1 Functions of Clock Output/Buzzer Output Controller 355
9.2 Configuration of Clock Output/Buzzer Output Controller 356
9.3 Registers Controlling Clock Output/Buzzer Output Controller 356
9.4 Operations of Clock Output/Buzzer Output Controller 359
9.4.1 Operation as output pin 359

CHAPTER 10 A/D CONVERTER 360

10.1 Function of A/D Converter 360
10.2 Configuration of A/D Converter 361
10.3 Registers Used in A/D Converter 363
10.4 A/D Converter Operations 373
10.4.1 Basic operations of A/D converter 373
10.4.2 Input voltage and conversion results 375
10.4.3 A/D converter operation modes 376
10.5 How to Read A/D Converter Characteristics Table 379
10.6 Cautions for A/D Converter 381

CHAPTER 11 SERIAL ARRAY UNIT 385

11.1 Functions of Serial Array Unit 385
11.1.1 3-wire serial I/O (CSI00, CSI01, CSI10, CSI20) 385
11.1.2 UART (UART0, UART1, UART2) 386
11.1.3 Simplified I^2C (IIC10, IIC20) 387
11.2 Configuration of Serial Array Unit 388
11.3 Registers Controlling Serial Array Unit 393
11.4 Operation Stop Mode 414
11.4.1 Stopping the operation by units 414
11.4.2 Stopping the operation by channels 415
11.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) Communication 417
11.5.1 Master transmission 418
11.5.2 Master reception 428
11.5.3 Master transmission/reception 437

11.5.4 Slave transmission 446
11.5.5 Slave reception 456
11.5.6 Slave transmission/reception 463
11.5.7 Calculating transfer clock frequency 473
11.5.8 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI20) communication 475
11.6 Operation of UART (UART0, UART1, UART2) Communication 476
11.6.1 UART transmission 477
11.6.2 UART reception 487
11.6.3 Calculating baud rate 494
11.6.4 Procedure for processing errors that occurred during UART (UART0, UART1, UART2) communication 498
11.7 Operation of Simplified I²C (IIC10, IIC20) Communication 499
11.7.1 Address field transmission 500
11.7.2 Data transmission 505
11.7.3 Data reception 509
11.7.4 Stop condition generation 514
11.7.5 Calculating transfer rate 515
11.7.6 Procedure for processing errors that occurred during simplified I²C (IIC10, IIC20) communication 518
11.8 Relationship Between Register Settings and Pins 519

CHAPTER 12 SERIAL INTERFACE IICA 525

12.1 Functions of Serial Interface IICA 525
12.2 Configuration of Serial Interface IICA 528
12.3 Registers Controlling Serial Interface IICA 531
12.4 I²C Bus Mode Functions 545
12.4.1 Pin configuration 545
12.4.2 Setting transfer clock by using IICWL and IICWH registers 546
12.5 I²C Bus Definitions and Control Methods 547
12.5.1 Start conditions 547
12.5.2 Addresses 548
12.5.3 Transfer direction specification 548
12.5.4 Acknowledge ($\overline{\text{ACK}}$) 549
12.5.5 Stop condition 550
12.5.6 Wait 551
12.5.7 Canceling wait 553
12.5.8 Interrupt request (INTIICA) generation timing and wait control 554
12.5.9 Address match detection method 555
12.5.10 Error detection 555
12.5.11 Extension code 555
12.5.12 Arbitration 556
12.5.13 Wakeup function 558
12.5.14 Communication reservation 561

12.5.15 Cautions ... 565
12.5.16 Communication operations ... 566
12.5.17 Timing of I²C interrupt request (INTIICA) occurrence ... 574
12.6 Timing Charts ... 595

CHAPTER 13 CEC TRANSMISSION/RECEPTION CIRCUIT ... 610

13.1 Functions of CEC Transmission/Reception Circuit ... 610
13.2 Configuration of CEC Transmission/Reception Circuit ... 613
13.3 Term Description ... 614
13.4 Registers Controlling Transmission/Reception Circuit ... 615
13.5 Start Bit and Data Bit Registers ... 632
13.5.1 Specifying low-level width and bit width of CEC transmission data ... 632
13.5.2 Checking CEC reception data timing ... 634
13.5.3 Determining CEC reception data logic (1 or 0) ... 637
13.5.4 Specifying bit width for detecting errors, signal-free time, and bus locking ... 638
13.6 Operation of CEC Transmission/Reception Circuit ... 639
13.6.1 CEC transmission/reception data format ... 639
13.6.2 Communication types ... 639
13.6.3 Bit timing ... 640
13.6.4 Header/data block ... 641
13.6.5 EOM (End of Message) ... 641
13.6.6 ACK (Acknowledge) ... 642
13.7 CEC Communication Functions ... 643
13.7.1 Communication bit width adjustment function ... 643
13.7.2 Receive bit timing check function ... 644
13.7.3 Initial CEC communication settings ... 645
13.7.4 CEC transmission ... 648
13.7.5 CEC reception ... 651
13.7.6 Error detection function ... 658
13.7.7 Clearing error flag ... 667
13.7.8 Signal-free time ... 668

CHAPTER 14 REMOTE CONTROLLER RECEIVER ... 670

14.1 Remote Controller Receiver Functions ... 670
14.2 Remote Controller Receiver Configuration ... 671
14.3 Registers to Control Remote Controller Receiver ... 679
14.4 Compare Registers of Remote Controller Receiver ... 686
14.5 Operation of Remote Controller Receiver ... 705
14.5.1 Format of type A reception mode ... 706
14.5.2 Operation flow of type A reception mode ... 706
14.5.3 Format of type B reception mode ... 708
14.5.4 Operation flow of type B reception mode ... 708
14.5.5 Format of type C reception mode ... 710

14.5.6 Operation flow of type C reception mode 710
14.5.7 Format of type B1 reception mode 712
14.5.8 Operation flow of type B1 reception mode 712
14.5.9 Format of type C1 reception mode 714
14.5.10 Operation flow of type C1 reception mode 714
14.5.11 Timing 716
14.5.12 Compare register setting 725
14.5.13 Error interrupt generation timing 728
14.5.14 Instruction for interrupt use 734

CHAPTER 15 MULTIPLIER/DIVIDER 739

15.1 Functions of Multiplier/Divider 739
15.2 Configuration of Multiplier/Divider 739
15.3 Register Controlling Multiplier/Divider 744
15.4 Operations of Multiplier/Divider 745
15.4.1 Multiplication operation 745
15.4.2 Division operation 746

CHAPTER 16 DMA CONTROLLER 748

16.1 Functions of DMA Controller 748
16.2 Configuration of DMA Controller 749
16.3 Registers Controlling DMA Controller 752
16.4 Operation of DMA Controller 755
16.4.1 Operation procedure 755
16.4.2 Transfer mode 756
16.4.3 Termination of DMA transfer 756
16.5 Example of Setting of DMA Controller 757
16.5.1 CSI consecutive transmission 757
16.5.2 Consecutive capturing of A/D conversion results 759
16.5.3 UART consecutive reception + ACK transmission 761
16.5.4 Holding DMA transfer pending by DWAITn 762
16.5.5 Forced termination by software 763
16.6 Cautions on Using DMA Controller 765

CHAPTER 17 INTERRUPT FUNCTIONS 767

17.1 Interrupt Function Types 767
17.2 Interrupt Sources and Configuration 767
17.3 Registers Controlling Interrupt Functions 773
17.4 Interrupt Servicing Operations 783
17.4.1 Maskable interrupt acknowledgment 783
17.4.2 Software interrupt request acknowledgment 785
17.4.3 Multiple interrupt servicing 786
17.4.4 Interrupt request hold 790

CHAPTER 18 KEY INTERRUPT FUNCTION 791

18.1 Functions of Key Interrupt 791
18.2 Configuration of Key Interrupt 791
18.3 Register Controlling Key Interrupt 792

CHAPTER 19 STANDBY FUNCTION 793

19.1 Standby Function and Configuration 793
19.1.1 Standby function 793
19.1.2 Registers controlling standby function 794
19.2 Standby Function Operation 797
19.2.1 HALT mode 797
19.2.2 STOP mode 802

CHAPTER 20 RESET FUNCTION 807

20.1 Register for Confirming Reset Source 818

CHAPTER 21 POWER-ON-CLEAR CIRCUIT 819

21.1 Functions of Power-on-Clear Circuit 819
21.2 Configuration of Power-on-Clear Circuit 820
21.3 Operation of Power-on-Clear Circuit 820
21.4 Cautions for Power-on-Clear Circuit 823

CHAPTER 22 LOW-VOLTAGE DETECTOR 825

22.1 Functions of Low-Voltage Detector 825
22.2 Configuration of Low-Voltage Detector 826
22.3 Registers Controlling Low-Voltage Detector 826
22.4 Operation of Low-Voltage Detector 830
22.4.1 When used as reset 831
22.4.2 When used as interrupt 837
22.5 Cautions for Low-Voltage Detector 843

CHAPTER 23 REGULATOR 847

23.1 Regulator Overview 847
23.2 Registers Controlling Regulator 847

CHAPTER 24 OPTION BYTE 849

24.1 Functions of Option Bytes 849
24.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H) 849
24.1.2 On-chip debug option byte (000C3H/ 010C3H) 850
24.2 Format of User Option Byte 850
24.3 Format of On-chip Debug Option Byte 852
24.4 Setting of Option Byte 853

CHAPTER 25 FLASH MEMORY 854

25.1 Writing with Flash Memory Programmer 854
25.2 Programming Environment 858
25.3 Communication Mode 858
25.4 Connection of Pins on Board 860
25.4.1 FLMD0 pin 860
25.4.2 TOOL0 pin 861
25.4.3 $\overline{\text{RESET}}$ pin 861
25.4.4 Port pins 861
25.4.5 REGC pin 861
25.4.6 X1 and X2 pins 861
25.4.7 Power supply 862
25.5 Registers that Control Flash Memory 862
25.6 Programming Method 863
25.6.1 Controlling flash memory 863
25.6.2 Flash memory programming mode 863
25.6.3 Selecting communication mode 864
25.6.4 Communication commands 865
25.7 Security Settings 866
25.8 Flash Memory Programming by Self-Programming 868
25.8.1 Boot swap function 870
25.8.2 Flash shield window function 872

CHAPTER 26 ON-CHIP DEBUG FUNCTION 873

26.1 Connecting QB-MINI2 to 78K0R/Kx3-C 873
26.2 On-Chip Debug Security ID 874
26.3 Securing of User Resources 874

CHAPTER 27 BCD CORRECTION CIRCUIT 876

27.1 BCD Correction Circuit Function 876
27.2 Registers Used by BCD Correction Circuit 876
27.3 BCD Correction Circuit Operation 877

CHAPTER 28 INSTRUCTION SET 879

28.1 Conventions Used in Operation List 880
28.1.1 Operand identifiers and specification methods 880
28.1.2 Description of operation column 881
28.1.3 Description of flag operation column 882
28.1.4 PREFIX instruction 882
28.2 Operation List 883

CHAPTER 29 ELECTRICAL SPECIFICATIONS 900

CHAPTER 30 PACKAGE DRAWINGS 949

CHAPTER 31 RECOMMENDED SOLDERING CONDITIONS 951

APPENDIX A DEVELOPMENT TOOLS 952

A.1 Software Package 955
A.2 Language Processing Software 955
A.3 Control Software 956
A.4 Flash Memory Programming Tools 957
A.4.1 When using flash memory programmer PG-FP5 and FL-PR5 957
A.4.2 When using on-chip debug emulator with programming function QB-MINI2 957
A.5 Debugging Tools (Hardware) 958
A.5.1 When using in-circuit emulator 958
A.5.2 When using on-chip debug emulator with programming function QB-MINI2 959
A.6 Debugging Tools (Software) 959

APPENDIX B REVISION HISTORY 960

B.1 Major Revisions in This Edition 960
B.2 Revision History of Preceding Editions 960

CHAPTER 1 OUTLINE

1.1 Features

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

Flash ROM	RAM	78K0R/KF3-C	78K0R/KG3-C
		80 Pins	100 Pins
128 KB	8 KB[Note]	μPD78F1847A	μPD78F1849A
96 KB	6 KB	μPD78F1846A	μPD78F1848A

Note This is 7 KB when the self-programming function is used.

- On-chip internal high-speed oscillation clocks
  - 20 MHz Internal high-speed oscillation clock: 20 MHz (TYP.)
  - 8 MHz Internal high-speed oscillation clock: 8 MHz (TYP.)
- 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 dedicated internal low-speed oscillation clock)
- On-chip multiplier/divider (16 bits × 16 bits, 32 bits ÷ 32 bits)
- On-chip key interrupt function
- On-chip clock output/buzzer output controller
- On-chip BCD adjustment
- I/O ports
  78K0R/KF3-C: 71 (N-ch open drain: 4)
  78K0R/KG3-C: 89 (N-ch open drain: 4)
- Timer: 13 channels
  - 16-bit timer: 11 channels
  - Watchdog timer: 1 channel
  - Real-time counter: 1 channel
- Serial interface
  - CSI: 2 channels/UART: 1 channel
  - CSI: 1 channel/UART: 1 channel/simplified I^2C: 1 channel
  - CSI: 1 channel/UART: 1 channel/simplified I^2C: 1 channel
  - I^2C: 1 channel
- 10-bit resolution A/D converter (AV_{REF} = 2.7 to 5.5 V)
  78K0R/KF3-C: 12 channels
  78K0R/KG3-C: 16 channels
- CEC transmission/reception circuit

- Remote controller receiver
- Power supply voltage: V_{DD} = 2.7 to 5.5 V
- Operating ambient temperature: T_A = –40 to +85°C

1.2 Applications

- Digital audio visual equipment

1.3 Ordering Information

- **Flash memory version**

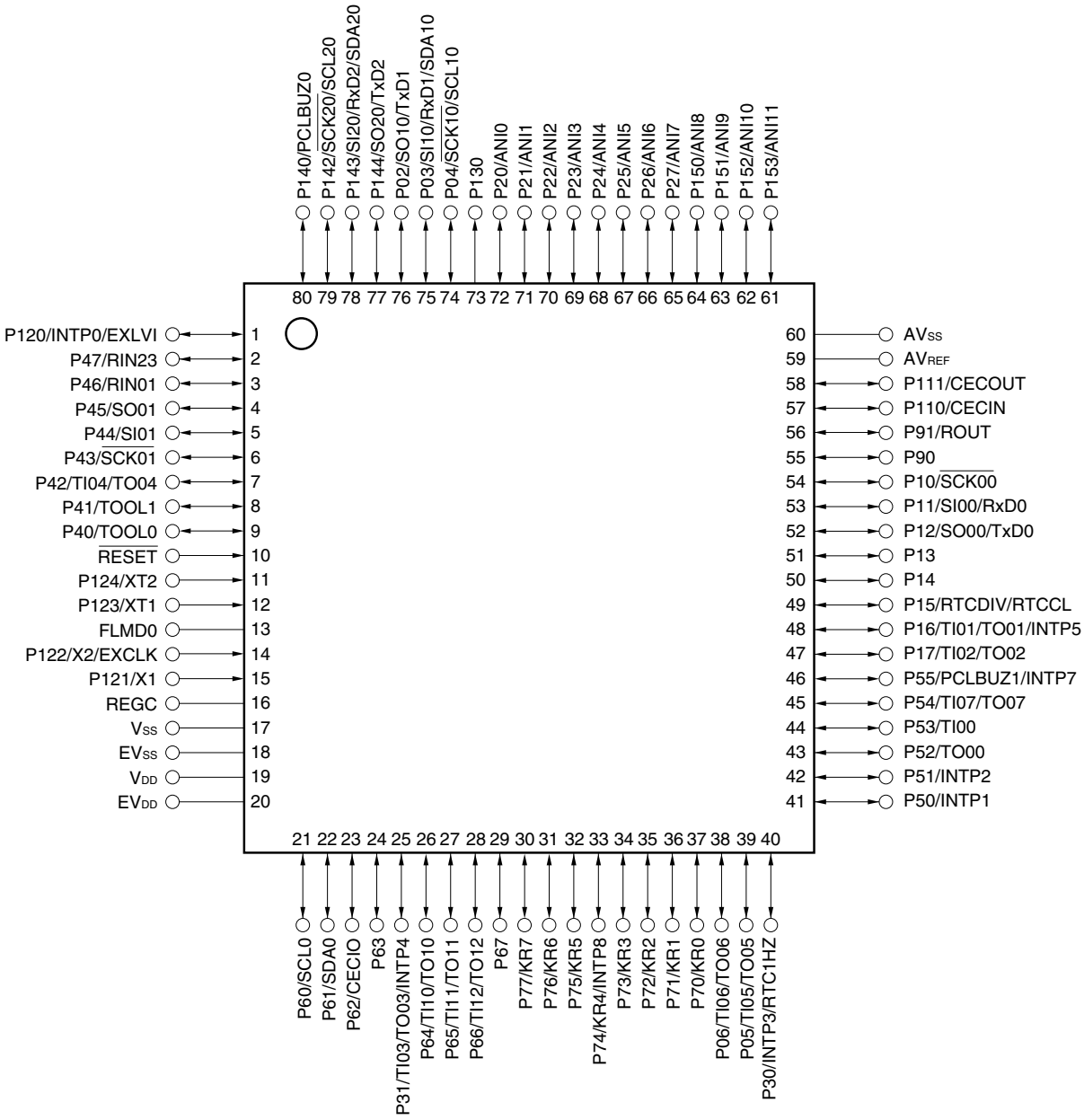
78K0R/Kx3-C Microcontroller	Package	Part Number
78K0R/KF3-C	80-pin plastic LQFP (fine pitch) (12 × 12)	μPD78F1846AGK-GAK-AX, 78F1847AGK-GAK-AX
78K0R/KG3-C	100-pin plastic LQFP (fine pitch) (14 × 14)	μPD78F1848AGC-UEU-AX, 78F1849AGC-UEU-AX

Caution The 78K0R/Kx3-C has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

1.4 Pin Configuration (Top View)

1.4.1 78K0R/KF3-C

- 80-pin plastic LQFP (fine pitch) (12 × 12)



Cautions 1. Make AVSS and EVSS the same potential as VSS.

2. Make EVDD the same potential as VDD.

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

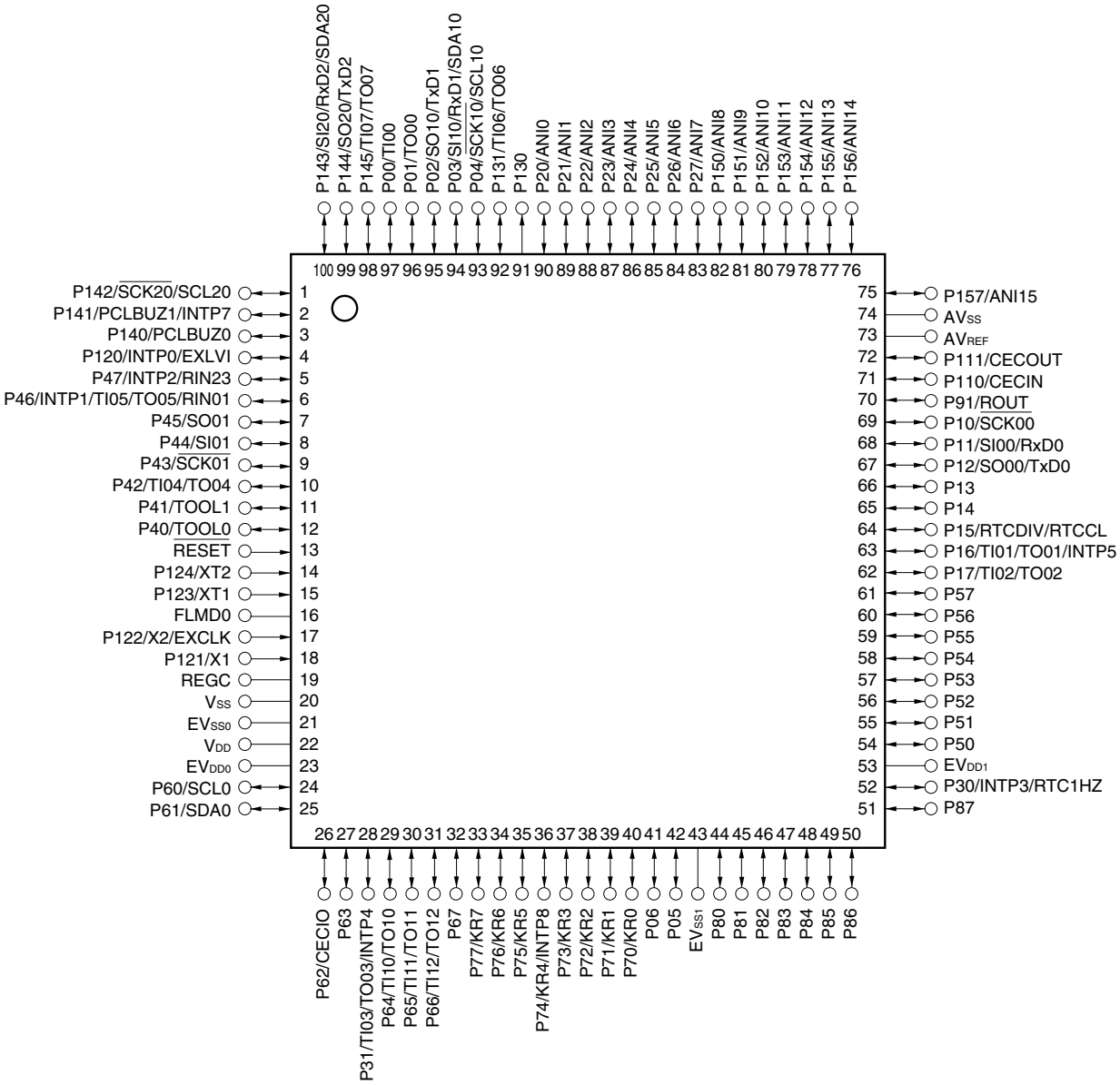
4. P20/ANI0 to P27/ANI7 and P150/ANI8 to P153/ANI11 are set as analog inputs in the order of P153/ANI11, …, P150/ANI8, P27/ANI7, …, P20/ANI0 by the A/D port configuration register (ADPC). When using P20/ANI0 to P27/ANI7 and P150/ANI8 to P153/ANI11 as analog inputs, start designing from P153/ANI11 (see 10.3 (6) A/D port configuration register (ADPC) for details).

Pin Identification

ANI0 to ANI11:	Analog input	PCLBUZ0, PCLBUZ1:	Programmable clock output/buzzer output
AV_{REF}:	Analog reference voltage		
AV_{SS}:	Analog ground	REGC:	Regulator capacitance
CECIN:	Consumer electronics control input	$\overline{RESET}$:	Reset
		RIN01:	Remote control input
CECIO:	Consumer electronics control input/output	RIN23:	Remote control input
		ROUT:	Remote control output
CECOUT:	Consumer electronics control output	RTC1HZ:	Real-time counter correction clock (1 Hz) output
EV_{DD}:	Power supply for port	RTCCL:	Real-time counter clock (32 kHz original oscillation) output
EV_{SS}:	Ground for port		
EXCLK:	External clock input (main system clock)	RTCDIV:	Real-time counter clock (32 kHz divided frequency) output
EXLVI:	External potential input for low-voltage detector	RxD0 to RxD2:	Receive data
		$\overline{SCK00}$, $\overline{SCK01}$,	
FLMD0:	Flash programming mode	$\overline{SCK10}$, $\overline{SCK20}$:	Serial clock input/output
INTP0 to INTP5,		SCL0, SCL10, SCL20:	Serial clock input/output
INTP7, INTP8:	External interrupt input	SDA0, SDA10, SDA20:	Serial data input/output
KR0 to KR7:	Key return	SI00, SI01,	
P02 to P06:	Port 0	SI10, SI20:	Serial data input
P10 to P17:	Port 1	SO00, SO01,	
P20 to P27:	Port 2	SO10, SO20:	Serial data output
P30, P31:	Port 3	TI00 to TI07, TI10 to TI12:	Timer input
P40 to P47:	Port 4	TO00 to TO07,	
P50 to P55:	Port 5	TO10 to TO12:	Timer output
P60 to P67:	Port 6	TOOL0:	Data input/output for tool
P70 to P77:	Port 7	TOOL1:	Clock output for tool
P90, P91:	Port 9	TxD0 to TxD2:	Transmit data
P110, P111:	Port 11	V_{DD}:	Power supply
P120 to P124:	Port 12	V_{SS}:	Ground
P130:	Port 13	X1, X2:	Crystal oscillator (main system clock)
P140, P142 to P144:	Port 14		
P150 to P153:	Port 15	XT1, XT2:	Crystal oscillator (subsystem clock)

1.4.2 78K0R/KG3-C

- 100-pin plastic LQFP (fine pitch) (14 × 14)



Cautions 1. Make AVSS, EVSS0, and EVSS1 the same potential as VSS.

2. Make EVDD0 and EVDD1 the same potential as VDD.

3. Connect the REGC pin to Vss 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 10.3 (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 VDD and two EVDD pins and connect the VSS and two EVSS pins to separate ground lines.

Pin Identification

ANI0 to ANI15:	Analog input
AV_{REF}:	Analog reference voltage
AV_{SS}:	Analog ground
CECIN:	Consumer electronics control input
CECIO:	Consumer electronics control input/output
CECOUT:	Consumer electronics control output
EV_{DD0}, EV_{DD1}:	Power supply for port
EV_{SS0}, EV_{SS1}:	Ground for port
EXCLK:	External clock input (main system clock)
EXLVI:	External potential input for low-voltage detector
FLMD0:	Flash programming mode
INTP0 to INTP5, INTP7, INTP8:	External interrupt input
KR0 to KR7:	Key return
P00 to P06:	Port 0
P10 to P17:	Port 1
P20 to P27:	Port 2
P30, P31:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P60 to P67:	Port 6
P70 to P77:	Port 7
P80 to P87:	Port 8
P91:	Port 9
P110, P111:	Port 11
P120 to P124:	Port 12
P130, P131:	Port 13
P140 to P145:	Port 14
P150 to P157:	Port 15
PCLBUZ0, PCLBUZ1:	Programmable clock output/buzzer output
REGC:	Regulator capacitance
$\overline{\text{RESET}}$:	Reset
RIN01:	Remote control input
RIN23:	Remote control input
ROUT:	Remote control output
RTC1HZ:	Real-time counter correction clock (1 Hz) output
RTCCL:	Real-time counter clock (32 kHz original oscillation) output
RTCDIV:	Real-time counter clock (32 kHz divided frequency) output
RxD0 to RxD2:	Receive data
$\overline{\text{SCK00}}$, $\overline{\text{SCK01}}$, $\overline{\text{SCK10}}$, $\overline{\text{SCK20}}$:	Serial clock input/output
SCL0, SCL10, SCL20:	Serial clock input/output
SDA0, SDA10, SDA20:	Serial data input/output
SI00, SI01, SI10, SI20:	Serial data input
SO00, SO01, SO10, SO20:	Serial data output
TI00 to TI07, TI10 to TI12:	Timer input
TO00 to TO07, TO10 to TO12:	Timer output
TOOL0:	Data input/output for tool
TOOL1:	Clock output for tool
TxD0 to TxD2:	Transmit data
V_{DD}:	Power supply
V_{SS}:	Ground
X1, X2:	Crystal oscillator (main system clock)
XT1, XT2:	Crystal oscillator (subsystem clock)

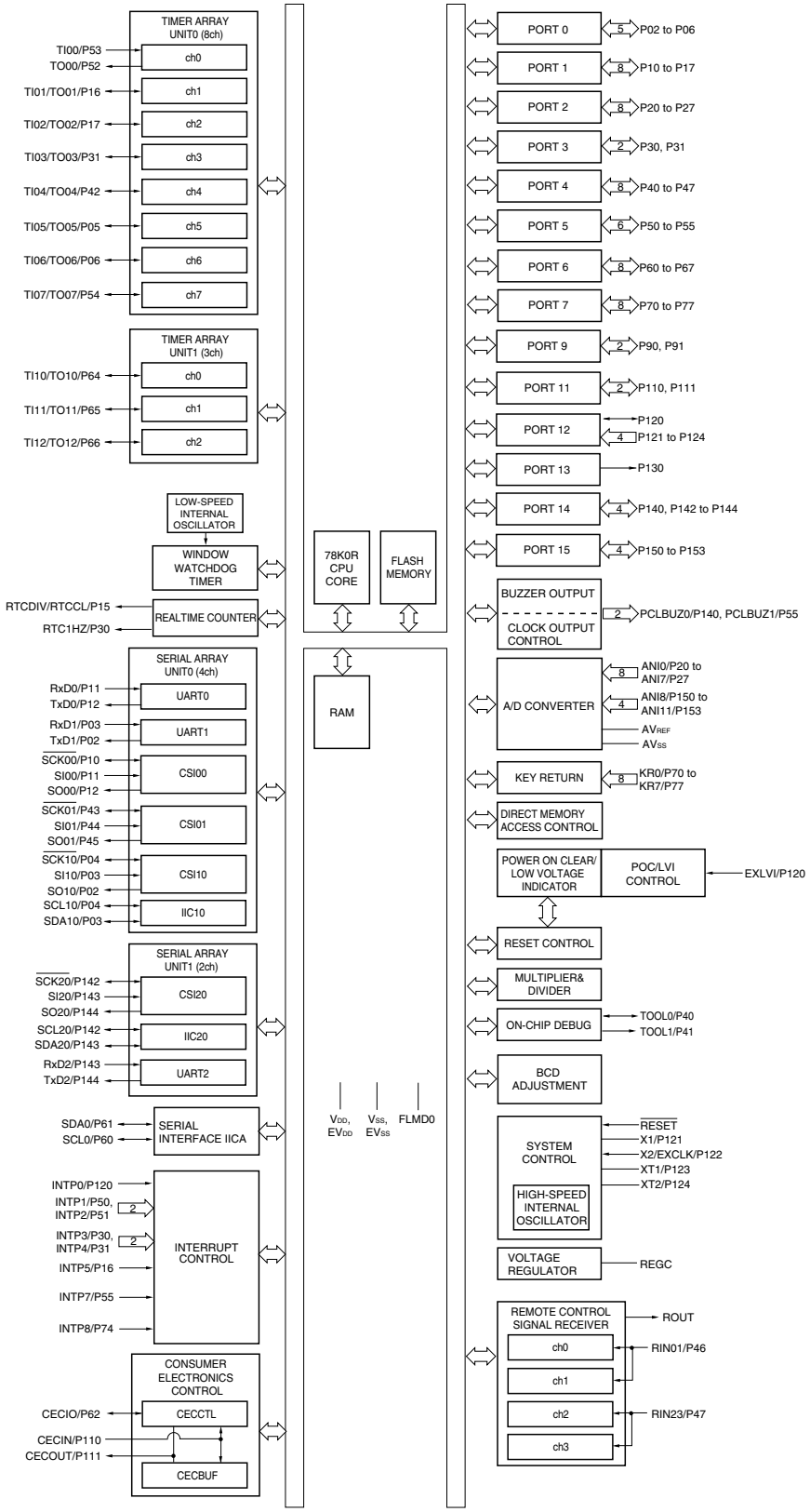
1.5 78K0R/Kx3-C Microcontroller Lineup

ROM	RAM	78K0R/KF3-C	78K0R/KG3-C
		80 Pins	100 Pins
128 KB	8 KB[Note]	μPD78F1847A	μPD78F1849A
96 KB	6 KB	μPD78F1846A	μPD78F1848A

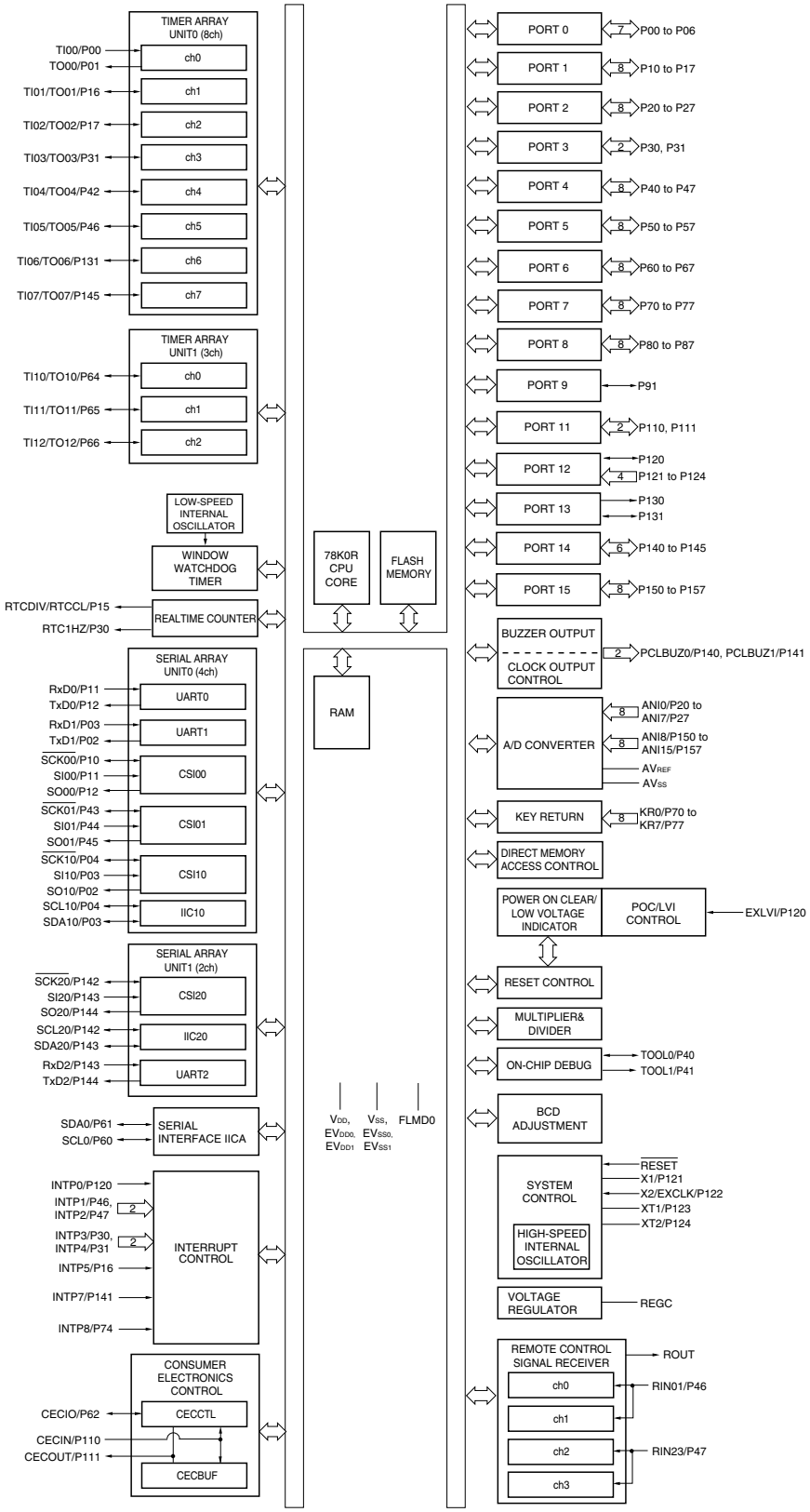
Note This is 7 KB when the self-programming function is used.

1.6 Block Diagram

1.6.1 78K0R/KF3-C



1.6.2 78K0R/KG3-C



1.7 Outline of Functions

(1/2)

Item		78K0R/KF3-C		78K0R/KG3-C	
		μPD78F1846	μPD78F1847	μPD78F1848	μPD78F1849
Internal memory	Flash memory (self-programming supported)	96 KB	128 KB	96 KB	128 KB
	RAM	6 KB	8 KB Note 1	6 KB	8 KB Note 1
Memory space		1 MB			
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			
	Internal high-speed oscillation clock	Internal oscillation 8 MHz (TYP.) : V_{DD} = 2.7 to 5.5 V			
	20 MHz internal high-speed oscillation clock	Internal oscillation 20 MHz (TYP.) : V_{DD} = 2.7 to 5.5 V			
Subsystem clock (Oscillation frequency)		XT1 (crystal) oscillation 32.768 kHz (TYP.): V_{DD} = 2.7 to 5.5 V			
	For CEC	65.536 kHz (TYP.): V_{DD} = 2.7 to 5.5 V			
Internal low-speed oscillation clock (dedicated to WDT)		Internal oscillation 30 kHz (TYP.): V_{DD} = 2.7 to 5.5 V			
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)			
Minimum instruction execution time		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)			
		61 μs (Subsystem clock: f_{SUB} = 32.768 kHz operation)			
Instruction set		• 8-bit operation, 16-bit operation • Multiplication (8 bits × 8 bits) • Bit manipulation (Set, reset, test, and Boolean operation), etc.			
I/O port	Total	71		89	
	CMOS I/O	62		80	
	CMOS input	4		4	
	CMOS output	1		1	
	N-ch open-drain I/O (6 V tolerance)	4		4	
Timer		• 16-bit timer: 11 channels (unit 0: 8 channels, unit 1: 3 channels) • Watchdog timer: 1 channel • Real-time counter (RTC): 1 channel			
	Timer output	11 (PWM outputs: timer array unit 0: 7, timer array unit 1: 2 Note 2)			
	RTC output	2 • 1 Hz (subsystem clock: f_{SUB} = 32.768 kHz) • 512 Hz, 16.384 kHz, or 32.768 kHz (subsystem clock: f_{SUB} = 32.768 kHz)			

Notes 1. This is 7 KB when the self-programming function is used.

2. The number of outputs varies, depending on the setting.

(2/2)

Item		78K0R/KF3-C		78K0R/KG3-C	
		μPD78F1846	μPD78F1847	μPD78F1848	μPD78F1849
Clock output/buzzer output		2 • 2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (peripheral hardware clock: f_{MAIN} = 20 MHz operation) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (Subsystem clock: f_{SUB} = 32.768 kHz operation)			
10-bit resolution A/D converter (AV_{REF} = 2.7 to 5.5 V)		12 channels		16 channels	
Serial interface		• CSI: 2 channels/UART: 1 channel • CSI: 1 channel/UART: 1 channel/simplified I^2C: 1 channel • CSI: 1 channel/UART: 1 channel/simplified I^2C: 1 channel • I^2C bus: 1 channel			
CEC transmission/reception circuit		Provided			
Remote controller receiver		Provided			
Multiplier/divider		• 16 bits × 16 bits = 32 bits (multiplication) • 32 bits ÷ 32 bits = 32 bits (division)			
DMA controller		2 channels			
Vectored interrupt sources	Internal	39			
	External	9			
Key interrupt		Key interrupt (INTKR) occurs by detecting falling edge of the key input pins (KR0 to KR7).			
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector • Internal reset by illegal instruction execution Note • Internal reset by a reset processing check error			
Power-on-clear circuit		• Power-on-reset: 1.61 ±0.09 V • Power-down-reset: 1.59 ±0.09 V			
Low-voltage detector		2.84 V to 4.22 V (10 stages)			
On-chip debug function		Provided			
Power supply voltage		V_{DD} = 2.7 to 5.5 V			
Operating ambient temperature		T_A = –40 to +85 °C			
Package		80-pin plastic LQFP (fine pitch) (12 × 12) (0.5 mm pitch)		100-pin plastic LQFP (fine pitch) (14 × 14) (0.5 mm pitch)	

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

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

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

Table 2-1. Pin I/O Buffer Power Supplies (AV_{REF}, EV_{DD}, V_{DD})

- 78K0R/KF3-C: 80-pin plastic LQFP (fine pitch) (12x12)

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27, P150 to P153
EV_{DD}	• Port pins other than P20 to P27, P121 to P124, and P150 to P153 • $\overline{RESET}$ and FLMD0 pins
V_{DD}	• P121 to P124 • Pins other than port pins (excluding $\overline{RESET}$ and FLMD0 pins)

Table 2-2. Pin I/O Buffer Power Supplies (AV_{REF}, EV_{DD0}, EV_{DD1}, V_{DD})

- 78K0R/KG3-C: 100-pin plastic LQFP (fine pitch) (14x14)

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27, P150 to P157
EV_{DD0}, EV_{DD1}	• Port pins other than P20 to P27, P121 to P124, and P150 to P157 • $\overline{RESET}$ and FLMD0 pins
V_{DD}	• P121 to P124 • Pins other than port pins (excluding $\overline{RESET}$ and FLMD0 pins)

2.1.1 78K0R/KF3-C

(1) Port functions (1/2): 78K0R/KF3-C

Function Name	I/O	Function	After Reset	Alternate Function
P02	I/O	Port 0. 5-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	SO10/TxD1
P03				SI10/RxD1/SDA10
P04				$\overline{SCK10}$/SCL10
P05				TI05/TO05
P06				TI06/TO06
P10	I/O	Port 1. 8-bit I/O port. Input of P10 and P11 can be set to TTL input buffer. Output of P10 and P12 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{SCK00}$
P11				SI00/RxD0
P12				SO00/TxD0
P13				–
P14				–
P15				RTCDIV/RTCCL
P16				TI01/TO01/INTP5
P17				TI02/TO02
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. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	RTC1HZ/INTP3
P31				TI03/TO03/INTP4
P40[Note]	I/O	Port 4. 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	TOOL0
P41				TOOL1
P42				TI04/TO04
P43				$\overline{SCK01}$
P44				SI01
P45				SO01
P46				RIN01
P47				RIN23
P50	I/O	Port 5. 6-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP1
P51				INTP2
P52				TO00
P53				TI00
P54				TI07/TO07
P55				PCLBUZ1/INTP7

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 (2/2): 78K0R/KF3-C

Function Name	I/O	Function	After Reset	Alternate Function
P60	I/O	Port 6. 8-bit I/O port. Input of P62 can be set to CEC input buffer. Output of P60 to P63 is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units. For only P62 and P64 to P67, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCL0
P61				SDA0
P62				CECIO
P63				–
P64				TI10/TO10
P65				TI11/TO11
P66				TI12/TO12
P67				–
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 to KR3
P74				KR4/INTP8
P75 to P77				KR5 to KR7
P90	I/O	Port 9. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	–
P91				ROUT
P110	I/O	Port 11. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	CECIN
P111				CECOUT
P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. For only P120, 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
P130	Output	Port 13. 1-bit output port.	Output port	–
P140	I/O	Port 14. 4-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
P142				$\overline{SCK20}$/SCL20
P143				SI20/RxD2/SDA20
P144				SO20/TxD2
P150 to P153	I/O	Port 15. 4-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI8 to ANI11

(2) Non-port functions (1/3): 78K0R/KF3-C

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 ANI11	Input	A/D converter analog input	Digital input port	P150 to P153
CECIN	Input	Serial data input for CEC	Input port	P110
CECIO	I/O	Serial data I/O for CEC	Input port	P62
CECOUT	Output	Serial data output for CEC	Input port	P111
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				P50
INTP2				P51
INTP3				P30/RTC1HZ
INTP4				P31/TI03/TO03
INTP5				P16/TI01/TO01
INTP7				P55/PCLBUZ1
INTP8				P74/KR4
KR0 to KR3	Input	Key interrupt input	Input port	P70 to P73
KR4				P74/INTP8
KR5 to KR7				P75 to P77
PCLBUZ0	Output	Clock output/buzzer output	Input port	P140
PCLBUZ1				P55/INTP7
REGC	–	Connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF).	–	–
RIN01	Input	Remote control receive data input (channels 0, 1)	Input port	P46
RIN23	Input	Remote control receive data input (channels 2, 3)	Input port	P47
ROUT	Output	Remote control receive data output	Input port	P91
RTCDIV	Output	Real-time counter clock (32 kHz divided frequency) output	Input port	P15/RTCCL
RTCCL	Output	Real-time counter clock (32 kHz original oscillation) output	Input port	P15/RTCDIV
RTC1HZ	Output	Real-time counter correction clock (1 Hz) output	Input port	P30/INTP3
$\overline{\text{RESET}}$	Input	System reset input	–	–
RxD0	Input	Serial data input to UART0	Input port	P11/SI00
RxD1	Input	Serial data input to UART1	Input port	P03/SI10/SDA10
RxD2	Input	Serial data input to UART2	Input port	P143/SI20/SDA20
$\overline{\text{SCK00}}$	I/O	Clock input/output for CSI00, CSI01, CSI10, and CSI20	Input port	P10
$\overline{\text{SCK01}}$				P43
$\overline{\text{SCK10}}$				P04/SCL10
$\overline{\text{SCK20}}$				P142/SCL20
SCL0	I/O	Clock input/output for I^2C	Input port	P60
SCL10	I/O	Clock input/output for simplified I^2C	Input port	P04/$\overline{\text{SCK10}}$
SCL20				P142/$\overline{\text{SCK20}}$

(2) Non-port functions (2/3): 78K0R/KF3-C

Function Name	I/O	Function	After Reset	Alternate Function
SDA0	I/O	Serial data I/O for I^2C	Input port	P61
SDA10		Serial data I/O for simplified I^2C	Input port	P03/SI10/RxD1
SDA20				P143/SI20/RxD2
SI00	Input	Serial data input to CSI00, CSI01, CSI10, and CSI20	Input port	P11/RxD0
SI01				P44
SI10				P03/RxD1/SDA10
SI20				P143/RxD2/SDA20
SO00	Output	Serial data output from CSI00, CSI01, CSI10, and CSI20	Input port	P12/TxD0
SO01				P45
SO10				P02/TxD1
SO20				P144/TxD2
TI00	Input	External count clock input to 16-bit timer 00	Input port	P53
TI01		External count clock input to 16-bit timer 01		P16/TO01/INTP5
TI02		External count clock input to 16-bit timer 02		P17/TO02
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		P05/TO05
TI06		External count clock input to 16-bit timer 06		P06/TO06
TI07		External count clock input to 16-bit timer 07		P54/TO07
TI10		External count clock input to 16-bit timer 10		P64/TO10
TI11		External count clock input to 16-bit timer 11		P65/TO11
TI12		External count clock input to 16-bit timer 12		P66/TO12
TO00	Output	16-bit timer 00 output	Input port	P52
TO01		16-bit timer 01 output		P16/TI01/INTP5
TO02		16-bit timer 02 output		P17/TI02
TO03		16-bit timer 03 output		P31/TI03/INTP4
TO04		16-bit timer 04 output		P42/TI04
TO05		16-bit timer 05 output		P05/TI05
TO06		16-bit timer 06 output		P06/TI06
TO07		16-bit timer 07 output		P54/TI07
TO10		16-bit timer 10 output		P64/TI10
TO11		16-bit timer 11 output		P65/TI11
TO12		16-bit timer 12 output		P66/TI12
TxD0	Output	Serial data output from UART0	Input port	P12/SO00
TxD1	Output	Serial data output from UART1	Input port	P02/SO10
TxD2	Output	Serial data output from UART2	Input port	P144/SO20
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

(2) Non-port functions (3/3): 78K0R/KF3-C

Function Name	I/O	Function	After Reset	Alternate Function
V_{DD}	–	Positive power supply (P121 to P124 and other than ports (excluding $\overline{RESET}$ and FLMD0 pins))	–	–
EV_{DD}	–	Positive power supply for ports (other than P20 to P27, P121 to P124, P150 to P153) and $\overline{RESET}$ and FLMD0 pins	–	–
AV_{REF}	–	• A/D converter reference voltage input • Positive power supply for P20 to P27, P150 to P153, and A/D converter	–	–
V_{SS}	–	Ground potential (P121 to P124 and other than ports (excluding $\overline{RESET}$ and FLMD0 pins))	–	–
EV_{SS}	–	Ground potential for ports (other than P20 to P27, P121 to P124, and P150 to P153) and $\overline{RESET}$ and FLMD0 pins	–	–
AV_{SS}	–	Ground potential for A/D converter, P20 to P27, and P150 to P153. Use this pin with the same potential as EV_{SS} and V_{SS}.	–	–
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

2.1.2 78K0R/KG3-C

(1) Port functions (1/2): 78K0R/KG3-C

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 7-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				$\overline{\text{SCK10}}$/SCL10
P05				–
P06				–
P10	I/O	Port 1. 8-bit I/O port. Input of P10 and P11 can be set to TTL input buffer. Output of P10 and P12 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{\text{SCK00}}$
P11				SI00/RxD0
P12				SO00/TxD0
P13				–
P14				–
P15				RTCDIV/RTCCL
P16				TI01/TO01/INTP5
P17				TI02/TO02
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. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	RTC1HZ/INTP3
P31				TI03/TO03/INTP4
P40[Note]	I/O	Port 4. 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	TOOL0
P41				TOOL1
P42				TI04/TO04
P43				$\overline{\text{SCK01}}$
P44				SI01
P45				SO01
P46				INTP1/TI05/TO05/RIN01
P47				INTP2/RIN23
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	–

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 (2/2): 78K0R/KG3-C

Function Name	I/O	Function	After Reset	Alternate Function
P60	I/O	Port 6. 8-bit I/O port. Input of P62 can be set to CEC input buffer. Output of P60 to P63 is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units. For only P62 and P64 to P67, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCL0
P61				SDA0
P62				CECIO
P63				–
P64				TI10/TO10
P65				TI11/TO11
P66				TI12/TO12
P67				–
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 to KR3
P74				KR4/INTP8
P75 to P77				KR5 to KR7
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	–
P91	I/O	Port 9. 1-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	ROUT
P110	I/O	Port 11. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	CECIN
P111				CECOUT
P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. For only P120, 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
P130	Output	Port 13. 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.	Output port	–
P131	I/O		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
P141				PCLBUZ1/INTP7
P142				$\overline{SCK20}$/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

(2) Non-port functions (1/3): 78K0R/KG3-C

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
CECIN	Input	Serial data input for CEC	Input port	P110
CECIO	I/O	Serial data I/O for CEC	Input port	P62
CECOUT	Output	Serial data output for CEC	Input port	P111
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/RIN01
INTP2				P47/RIN23
INTP3				P30/RTC1HZ
INTP4				P31/TI03/TO03
INTP5				P16/TI01/TO01
INTP7				P141/PCLBUZ1
INTP8				P74/KR4
KR0 to KR3	Input	Key interrupt input	Input port	P70 to P73
KR4				P74/INTP8
KR5 to KR7				P75 to P77
PCLBUZ0	Output	Clock output/buzzer output	Input port	P140
PCLBUZ1				P141/INTP7
REGC	–	Connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF).	–	–
RIN01	Input	Remote control receive data input (channels 0, 1)	Input port	P46/INTP1/TI05/TO05
RIN23	Input	Remote control receive data input (channels 2, 3)	Input port	P47/INTP2
ROUT	Output	Remote control receive data output	Input port	P91
RTCDIV	Output	Real-time counter clock (32 kHz divided frequency) output	Input port	P15/RTCCL
RTCCL	Output	Real-time counter clock (32 kHz original oscillation) output	Input port	P15/RTCDIV
RTC1HZ	Output	Real-time counter correction clock (1 Hz) output	Input port	P30/INTP3
$\overline{\text{RESET}}$	Input	System reset input	–	–
RxD0	Input	Serial data input to UART0	Input port	P11/SI00
RxD1	Input	Serial data input to UART1	Input port	P03/SI10/SDA10
RxD2	Input	Serial data input to UART2	Input port	P143/SI20/SDA20
$\overline{\text{SCK00}}$	I/O	Clock input/output for CSI00, CSI01, CSI10, and CSI20	Input port	P10
$\overline{\text{SCK01}}$				P43
$\overline{\text{SCK10}}$				P04/SCL10
$\overline{\text{SCK20}}$				P142/SCL20
SCL0	I/O	Clock input/output for I^2C	Input port	P60
SCL10	I/O	Clock input/output for simplified I^2C	Input port	P04/$\overline{\text{SCK10}}$
SCL20				P142/$\overline{\text{SCK20}}$

(2) Non-port functions (2/3): 78K0R/KG3-C

Function Name	I/O	Function	After Reset	Alternate Function
SDA0	I/O	Serial data I/O for I^2C	Input port	P61
SDA10	I/O	Serial data I/O for simplified I^2C	Input port	P03/SI10/RxD1
SDA20				P143/SI20/RxD2
SI00	Input	Serial data input to CSI00, CSI01, CSI10, and CSI20	Input port	P11/RxD0
SI01				P44
SI10				P03/RxD1/SDA10
SI20				P143/RxD2/SDA20
SO00	Output	Serial data output from CSI00, CSI01, CSI10, and CSI20	Input port	P12/TxD0
SO01				P45
SO10				P02/TxD1
SO20				P144/TxD2
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
TI02		External count clock input to 16-bit timer 02		P17/TO02
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/ RIN01
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		P64/TO10
TI11		External count clock input to 16-bit timer 11		P65/TO11
TI12		External count clock input to 16-bit timer 12		P66/TO12
TO00	Output	16-bit timer 00 output	Input port	P01
TO01		16-bit timer 01 output		P16/TI01/INTP5
TO02		16-bit timer 02 output		P17/TI02
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/RIN01
TO06		16-bit timer 06 output		P131/TI06
TO07		16-bit timer 07 output		P145/TI07
TO10		16-bit timer 10 output		P64/TI10
TO11		16-bit timer 11 output		P65/TI11
TO12		16-bit timer 12 output		P66/TI12
TxD0	Output	Serial data output from UART0	Input port	P12/SO00
TxD1	Output	Serial data output from UART1	Input port	P02/SO10
TxD2	Output	Serial data output from UART2	Input port	P144/SO20
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

(2) Non-port functions (3/3): 78K0R/KG3-C

Function Name	I/O	Function	After Reset	Alternate Function
V_{DD}	–	Positive power supply (P121 to P124 and other than ports (excluding $\overline{RESET}$ and FLMD0 pins))	–	–
EV_{DD0}, EV_{DD1}	–	Positive power supply for ports (other than P20 to P27, P121 to P124, P150 to P157) and $\overline{RESET}$ and FLMD0 pins	–	–
AV_{REF}	–	• A/D converter reference voltage input • Positive power supply for P20 to P27, P150 to P157, and A/D converter	–	–
V_{SS}	–	Ground potential (P121 to P124 and other than ports (excluding $\overline{RESET}$ and FLMD0 pins))	–	–
EV_{SS0}, EV_{SS1}	–	Ground potential for ports (other than P20 to P27, P121 to P124, and P150 to P157) and $\overline{RESET}$ and FLMD0 pins	–	–
AV_{SS}	–	Ground potential for A/D converter, P20 to P27, and P150 to P157. Use this pin with the same potential as EV_{SS0}, EV_{SS1}, and V_{SS}.	–	–
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

Remark The pins mounted depend on the product. See **1.4 Pin Configuration (Top View)** and **2.1 Pin Function List**.

2.2.1 P00 to P06 (port 0)

P00 to P06 function as an I/O port. These pins also function as timer I/O, serial interface data I/O, and clock I/O.

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).

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P00/TI00	–[Note 1]	√
P01/TO00	–[Note 1]	√
P02/SO10/TxD1	√	√
P03/SI10/RxD1/SDA10	√	√
P04/$\overline{\text{SCK10}}$/SCL10	√	√
P05/TI05/TO05	√	P05[Note 2]
P06/TI06/TO06	√	P06[Note 2]

Notes 1. TI00 and TO00 are shared with following pins, respectively, in the 78K0R/KF3-C.
P53/TI00, P52/TO00

2. The 78K0R/KG3-C does not have a sharing function.
TI05/TO05 and TI06/TO06 are shared with following pins, respectively, in the 78K0R/KG3-C.
P46/INTP1/TI05/TO05/RIN01, P131/TI06/TO06

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

(1) Port mode

P00 to P06 function as an I/O port. P00 to P06 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, and clock I/O.

(a) TI00, TI05, TI06

These are the pins for inputting an external count clock/capture trigger to 16-bit timer 00, 05, and 06.

(b) TO00, TO05, TO06

These are the timer output pins of 16-bit timer 00, 05, and 06.

(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^2C.

(i) SCL10

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

Caution 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 11-7 Relationship Between Register Settings and Pins (Channel 2 of Unit 0: CSI10, UART1 Transmission, IIC10)**
- **Table 11-8 Relationship Between Register Settings and Pins (Channel 3 of Unit 0: UART1 Reception)**

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, and real-time counter clock output.

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

Output from the P10 and P12 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 1 (POM1).

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 P12 and P15 to P17 function as external interrupt request input, serial interface data I/O, clock I/O, timer I/O, and real-time counter clock 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) TxD0

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

(f) TI01, TI02

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

(g) TO01, TO02

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

(h) 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.

(i) RTCDIV

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

(j) RTCCL

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

Cautions 1. To use P10/$\overline{\text{SCK00}}$, P11/SI00/RxD0, or P12/SO00/TxD0 as a general-purpose port, note the serial array unit setting. For details, refer to the following tables.

- **Table 11-5 Relationship Between Register Settings and Pins (Channel 0 of Unit 0: CSI00, UART0 Transmission)**
- **Table 11-6 Relationship Between Register Settings and Pins (Channel 1 of Unit 0: CSI01, UART0 Reception)**

In addition, clear port output mode register 1 (POM1) to 00H.

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 **10.6 (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, P31 (port 3)

P30 and P31 function as a 2-bit I/O port. These 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 and P31 function as a 2-bit I/O port. P30 and P31 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 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, timer I/O, and remote control receive data input.

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P40/TOOL0	√	√
P41/TOOL1	√	√
P42/TI04/TO04	√	√
P43/$\overline{\text{SCK01}}$	√	√
P44/SI01	√	√
P45/SO01	√	√
P46/INTP1/TI05/TO05/RIN01	P46/RIN01[Note]	√
P47/INTP2/RIN23	P47/RIN23[Note]	√

Note INTP1, TI05/TO05, and INTP2 are shared with following pins, respectively, in the 78K0R/KF3-C.
P50/INTP1, P05/TI05/TO05, P51/INTP2

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

(1) Port mode

P40 to P47 function as an 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, timer I/O, and remote control receive data input.

(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.

(i) RIN01

This is a remote control receive data input pin (channels 0, 1).

(j) RIN23

This is a remote control receive data input pin (channels 2, 3).

Caution 1. 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_{DD0} or EV_{DD1} 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_{DD0} or EV_{DD1} via an external resistor.

Caution 2. To use P43/$\overline{SCK01}$, P44/SI01, or P45/SO01 as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 11-6 Relationship Between Register Settings and Pins (Channel 1 of Unit 0: CSI01, UART0 reception).

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

2.2.6 P50 to P57 (port 5)

P50 to P57 function as an I/O port. These pins also function as external interrupt request input, timer I/O, and clock/buzzer output.

P50 to P57 can be set to input or output port in 1-bit units using port mode register 5 (PM5).

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P50/INTP1	√	P50[Note]
P51/INTP2	√	P51[Note]
P52/TO00	√	P52[Note]
P53/TI00	√	P53[Note]
P54/TI07/TO07	√	P54[Note]
P55/PCLBUZ1/INTP7	√	P55[Note]
P56	–	√
P57	–	√

Note The 78K0R/KG3-C does not have a sharing function.

INTP1, INTP2, TO00, TI00, TI07/TO07, and PCLBUZ1/INTP7/INTP7 are shared with following pins, respectively, in the 78K0R/KG3-C.

P46/INTP1/TI05/TO05/RIN01, P47/INTP2/RIN23, P01/TO00, P00/TI00, P145/TI07/TO07, P141/PCLBUZ1/INTP7

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

(1) Port mode

P50 to P57 function as an 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 interrupt request input, timer I/O, and clock/buzzer output.

(a) INTP1, INTP2, INTP7

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) TI00, TI07

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 00 and 07.

(c) TO00, TO07

These are the timer output pins from 16-bit timers 00 and 07.

(d) PCLBUZ1

This is a clock/buzzer output pins.

2.2.7 P60 to P67 (port 6)

P60 to P67 function as an 8-bit I/O port. These pins also function as serial interface data I/O, clock I/O, serial data I/O for CEC, and timer I/O.

Input to the P62 pin can be specified through a normal input buffer or a CEC input buffer, using port function register 6 (PF6) and port input mode register 6 (PIM6) (see **Figure 4-57**).

Only the P62 pin can be connected to an internal diode and pull-up resistor by setting port function register 6 (PF6) and pull-up resistor option register 6 (PU6) (see **Figure 4-57**).

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 P62 and 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 to P62 and P64 to P66 function as serial interface data I/O, clock I/O, serial data I/O for CEC, and timer I/O.

(a) SDA0

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

(b) SCL0

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

(c) CECIO

This is a serial data I/O pin for CEC.

(d) TI10 to TI12

These are the pins for inputting an external count clock/capture trigger to 16-bit timers 10 to 12.

(e) TO10 to TO12

These are the timer output pins of 16-bit timers 10 to 12.

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 and external interrupt request input.

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, and external interrupt request input.

(a) KR0 to KR7

These are the key interrupt input pins

(b) INTP8

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.

2.2.9 P80 to P87 (port 8) (78K0R/KG3-C only)

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).

2.2.10 P90, P91 (port 9)

P90 and P91 function as an I/O port. These pins also function as remote control receive data output.

By specifying a setting using the remote controller receive data through control register (RMSW), it is possible to output the remote controller receive data input from the RIN01 and RIN23 pins from these pins without noise elimination or decoding (For details, see 14.3 (5) Remote controller receive data through control register (RMSW)).

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P90	√	–
P91/ROUT	√	√

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

(1) Port mode

P90 and P91 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

P91 function as remote control receive data output.

(a) ROUT

This is a remote control receive data output pin.

2.2.11 P110, P111 (port 11)

P110 and P111 function as a 2-bit I/O port. These pins also function as serial data I/O for CEC.

The I/O port mode or CECIN mode can be specified for the P110 pin by using port function register 11 (PF11) (see **Figure 4-58**).

The I/O port mode or CECOUT mode can be specified for the P111 pin by using port function register 11 (PF11) (see **Figure 4-58**).

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

(1) Port mode

P110 and P111 function as a 2-bit I/O port. P110 and P111 can be set to input or output port in 1-bit units using port mode register 11 (PM11). 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 serial data I/O for CEC.

(a) CECIN

This is a serial data input pin for CEC.

(b) CECOUT

This is a serial data output pin for CEC.

2.2.12 P120 to P124 (port 12)

P120 function as a 1-bit I/O port. P121 to P124 functions 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, and external clock input for main system clock.

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

(1) Port mode

P120 functions as a 1-bit I/O port. P120 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 functions as a 4-bit input port.

(2) Control mode

P120 to P124 function as external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, and external clock input for main system clock.

(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.

2.2.13 P130, P131 (port 13)

P130 functions as an output port. P131 functions as an I/O port. These pins 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**).

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P130	√	√
P131/TI06/TO06	–Note	√

Note TI06/TO06 is shared with following pin, in the 78K0R/KF3-C.
P06/TI06/TO06

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

(1) Port mode

P130 functions as an output port.

P131 functions as an I/O port. P131 can be set to input or output port 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 an I/O port. These 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).

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P140/PCLBUZ0	√	√
P141/PCLBUZ1/INTP7	–Note	√
P142/$\overline{\text{SCK20}}$/SCL20	√	√
P143/SI20/RxD2/SDA20	√	√
P144/SO20/TxD2	√	√
P145/TI07/TO07	–Note	√

Note PCLBUZ1/INTP7 and TI07/TO07 are shared with following pins, respectively, in the 78K0R/KF3-C.
P55/PCLBUZ1/INTP7, P54/TI07/TO07

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

(1) Port mode

P140 to P145 function as an 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) INTP7

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) 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^2C.

(k) SCL20

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

Caution To use P142/$\overline{\text{SCK20}}$/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 11-9 Relationship Between Register Settings and Pins (Channel 0 of Unit 1: CSI20, UART2 Transmission, IIC20)**
- **Table 11-10 Relationship Between Register Settings and Pins (Channel 1 of Unit 1: UART2 Reception)**

In addition, clear port output mode register 14 (POM14) to 00H.

2.2.15 P150 to P157 (port 15)

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

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P150/ANI8	√	√
P151/ANI9	√	√
P152/ANI10	√	√
P153/ANI11	√	√
P154/ANI12	–	√
P155/ANI13	–	√
P156/ANI14	–	√
P157/ANI15	–	√

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

(1) Port mode

P150 to P157 function as an 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 **10.6 (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 AV_{REF}

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.

When all pins of ports 2 and 15 are used as the analog port pins, make the potential of AV_{REF} be such that 2.7 V ≤ $AV_{REF} \leq V_{DD}$. When one or more of the pins of ports 2 and 15 are used as the digital port pins or when the A/D converter is not used, make AV_{REF} the same potential as EV_{DD0}, EV_{DD1}, or V_{DD}.

2.2.17 AV_{SS}

This is the ground potential pin of A/D converter, P20 to P27, and P150 to P157. Even when the A/D converter is not used, always use this pin with the same potential as EV_{SS0}, EV_{SS1}, and V_{SS}.

Remarks 1. P20/ANI0 to P27/ANI7, P150/ANI8 to P153/ANI11: 78K0R/KF3-C
P20/ANI0 to P27/ANI7, P150/ANI8 to P157/ANI15: 78K0R/KG3-C

2. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

2.2.18 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

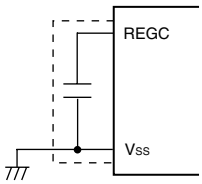
When the external reset pin is not used, connect this pin directly or via a resistor to EV_{DD0} or EV_{DD1}.

When the external reset pin is used, design the circuit based on V_{DD}.

2.2.19 REGC

This is the pin for connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF).

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



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

2.2.20 V_{DD}, EV_{DD0}, EV_{DD1}

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

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

2.2.21 V_{SS}, EV_{SS0}, EV_{SS1}

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

EV_{SS0} and EV_{SS1} are the ground potential pins for ports other than P20 to P27, P121 to P124, and P150 to P157 as well as for the $\overline{\text{RESET}}$ and FLMD0 pins.

Remarks 1. P150 to P153: 78K0R/KF3-C
P150 to P157: 78K0R/KG3-C

2. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

2.2.22 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_{SS} 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_{SS} 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 kΩ to 200 kΩ.

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_{DD} 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.

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

2.3.1 78K0R/KF3-C

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

Table 2-3. Connection of Unused Pins (78K0R/KF3-C) (1/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P02/SO10/TxD1	5-AG	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level output.
P03/SI10/RxD1/SDA10	5-AN		
P04/$\overline{SCK10}$/SCL10			
P05/TI05/TO05	8-R		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P06/TI06/TO06			
P10/$\overline{SCK00}$	5-AN		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level output.
P11/SI00/RxD0			Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P12/SO00/TxD0	5-AG		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level output.
P13			Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P14	8-R		
P15/RTCDIV/RTCCL	5-AG		
P16/TI01/TO01/INTP5	8-R		
P17/TI02/TO02			
P20/ANI0 to P27/ANI7[Note]	11-G		Input: Independently connect to AV_{REF} or AV_{SS} via a resistor. Output: Leave open.
P30/RTC1HZ/INTP3	8-R		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P31/TI03/TO03/INTP4			
P40/TOOL0			<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_{DD} or EV_{SS} via a resistor. Output: Leave open.
P41/TOOL1	5-AG		Input: Independently connect to EV_{DD0} or EV_{SS} via a resistor. Output: Leave open.
P42/TI04/TO04	8-R		
P43/$\overline{SCK01}$			
P44/SI01			
P45/SO01	5-AG		
P46/RIN01	8-R		
P47/RIN23			

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

Table 2-3. Connection of Unused Pins (78K0R/KF3-C) (2/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P50/INTP1	8-R	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P51/INTP2			
P52/TO00	5-AG		
P53/TI00	8-R		
P54/TI07/TO07			
P55/PCLBUZ1/INTP7			
P60/SCL0	13-R		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor, or connect directly to EV_{SS}. Output: Set the port output latch to 0 and leave these pins open via low-level output.
P61/SDA0			
P62/CECIO	13-AJ		
P63	13-P		
P64/TI10/TO10	8-R		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P65/TI11/TO11			
P66/TI12/TO12			
P67			
P70/KR0 to P73/KR3			
P74/KR4/INTP8			
P75/KR5 to P77/KR7			
P90	5-AG		
P91/ROUT			
P110/CECIN	8-R		
P110/CECOUT	5-AG		
P120/INTP0/EXLVI	8-R		
P121/X1[Note 1]	37-C	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK[Note 1]			
P123/XT1[Note 1]	37-B		
P124/XT2[Note 1]			
P130	3-C	Output	Leave open.
P140/PCLBUZ0	8-R	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P142/$\overline{SCK20}$/SCL20	5-AN		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P143/SI20/RxD2/SDA20			
P144/SO20/TxD2	5-AG		<When N-ch open-drain> Set the port output latch to 0 and leave open with low level out put.
P150/ANI8 to P153/ANI11[Note 2]	11-G		Input: Independently connect to AV_{REF} or AV_{SS} via a resistor. Output: Leave open.

Notes 1. Use recommended connection above in input port mode (see **Figure 5-2 Format of Clock Operation Mode Control Register (CMC)**) when these pins are not used.

2. P150/ANI8 to P153/ANI11 are set in the digital input port mode after release of reset.

Table 2-3. Connection of Unused Pins (78K0R/KF3-C) (3/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
AV_{REF}	–	–	<When one or more of P20 to P27 and P150 to P153 are set as a digital port> Make this pin the same potential as EV_{DD} or V_{DD}. <When all of P20 to P27 and P150 to P153 are set as analog ports> Make this pin to have a potential where 2.7 V ≤ AV_{REF} ≤ V_{DD}.
AV_{SS}	–	–	Make this pin the same potential as EV_{SS} or V_{SS}.
FLMD0	2-W	–	Leave open or connect to V_{SS} via a resistor of 100 kΩ or more.
$\overline{\text{RESET}}$	2	Input	Connect directly or via a resistor to EV_{DD}.
REGC	–	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).

2.3.2 78K0R/KG3-C

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 (78K0R/KG3-C) (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_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P01/TO00	5-AG		
P02/SO10/TxD1			Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level out put.
P03/SI10/RxD1/SDA10	5-AN		
P04/$\overline{SCK10}$/SCL10			
P05, P06	8-R		Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P10/$\overline{SCK00}$	5-AN		Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level out put.
P11/SI00/RxD0			Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P12/SO00/TxD0	5-AG		Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open. <When N-ch open-drain> Set the port output latch to 0 and leave open with low level out put.
P13			Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P14	8-R		
P15/RTCDIV/RTCCL	5-AG		
P16/TI01/TO01/INTP5	8-R		
P17/TI02/TO02			
P20/ANI0 to P27/ANI7[Note]	11-G		Input: Independently connect to AV_{REF} or AV_{SS} via a resistor. Output: Leave open.

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 (78K0R/KG3-C) (2/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P30/RTC1HZ/INTP3	8-R	I/O	Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P31/TI03/TO03/INTP4			
P40/TOOL0			<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_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P41/TOOL1	5-AG		Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P42/TI04/TO04	8-R		
P43/$\overline{SCK01}$			
P44/SI01			
P45/SO01	5-AG		
P46/TI05/TO05/INTP1/RIN01	8-R		
P47/INTP2/RIN23			
P50, P51, P53 to P55	8-R		Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P52, P56, P57	5-AG		
P60/SCL0	13-R		Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor, or connect directly to EV_{SS0} or EV_{SS1}. Output: Set the port output latch to 0 and leave these pins open via low-level output.
P61/SDA0			
P62/CECIO	13-AJ		
P63	13-P		
P64/TI10/TO10	8-R		Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P65/TI11/TO11			
P66/TI12/TO12			
P67			
P70/KR0 to P73/KR3			
P74/KR4/INTP8			
P75/KR5 to P77/KR7			
P80 to P87	5-AG		
P91/ROUT			
P110/CECIN	8-R		
P110/CECOUT	5-AG		
P120/INTP0/EXLVI	8-R		
P121/X1[Note]	37-C	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK[Note]			
P123/XT1[Note]	37-B		
P124/XT2[Note]			
P130	3-C	Output	Leave open.

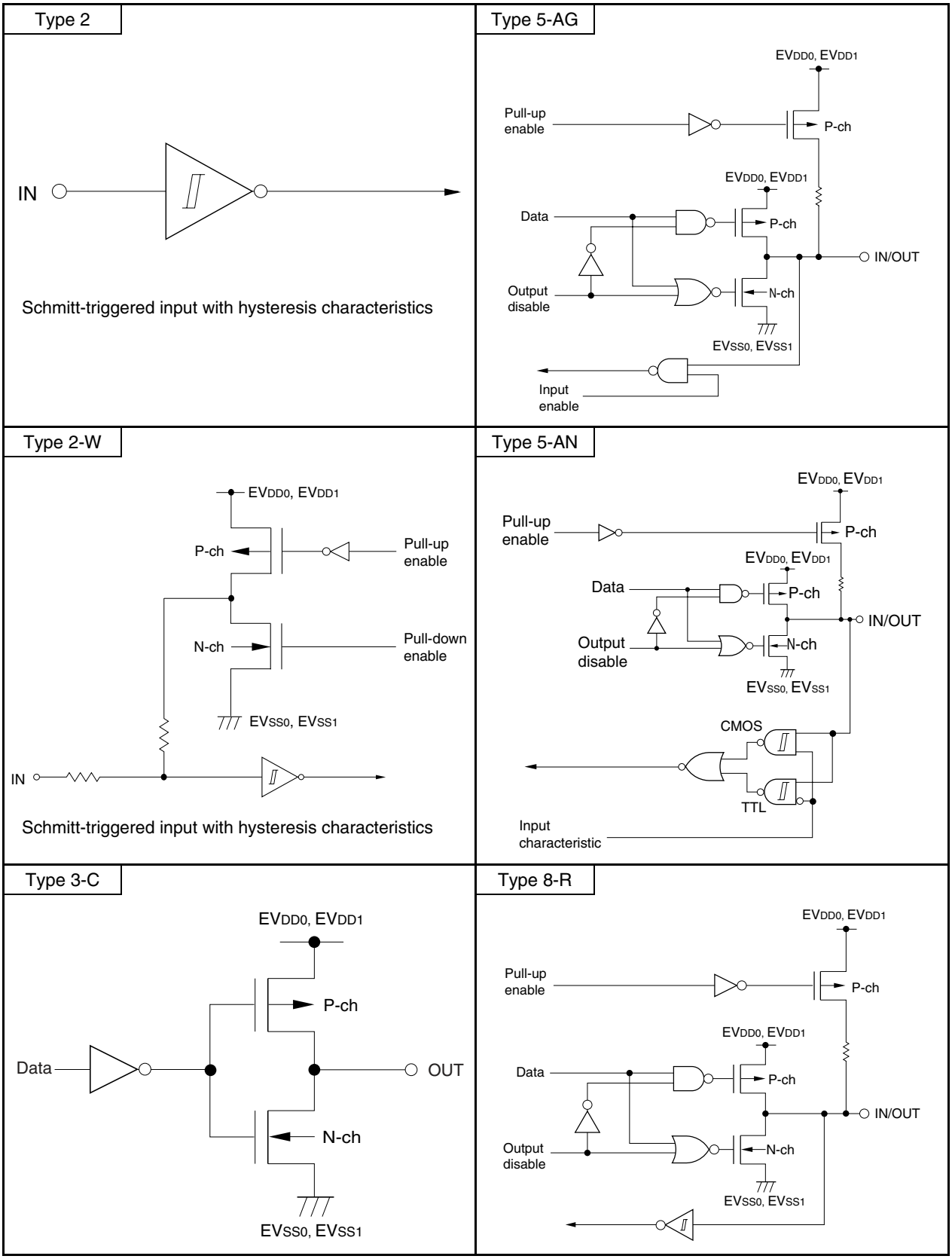
Note Use recommended connection above in input port mode (see **Figure 5-2 Format of Clock Operation Mode Control Register (CMC)**) when these pins are not used.

Table 2-4. Connection of Unused Pins (78K0R/KG3-C) (3/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P131/TI06/TO06	8-R	I/O	Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P140/PCLBUZ0			
P141/PCLBUZ1/INTP7			
P142/$\overline{SCK20}$/SCL20	5-AN		Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P143/SI20/RxD2/SDA20			
P144/SO20/TxD2	5-AG		<When N-ch open-drain> Set the port output latch to 0 and leave open with low level out put.
P145/TI07/TO07	8-R		Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} via a resistor. Output: Leave open.
P150/ANI8 to P157/ANI15[Note]	11-G		Input: Independently connect to AV_{REF} or AV_{SS} via a resistor. Output: Leave open.
AV_{REF}	–	–	<When one or more of P20 to P27 and P150 to P157 are set as a digital port> Make this pin the same potential as EV_{DD0}, EV_{DD1}, or V_{DD}. <When all of P20 to P27 and P150 to P157 are set as analog ports> Make this pin to have a potential where 2.7 V ≤ AV_{REF} ≤ V_{DD}.
AV_{SS}	–	–	Make this pin the same potential as EV_{SS0}, EV_{SS1}, or V_{SS}.
FLMD0	2-W	–	Leave open or connect to V_{SS} via a resistor of 100 kΩ or more.
$\overline{RESET}$	2	Input	Connect directly or via a resistor to EV_{DD0} or EV_{DD1}.
REGC	–	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).

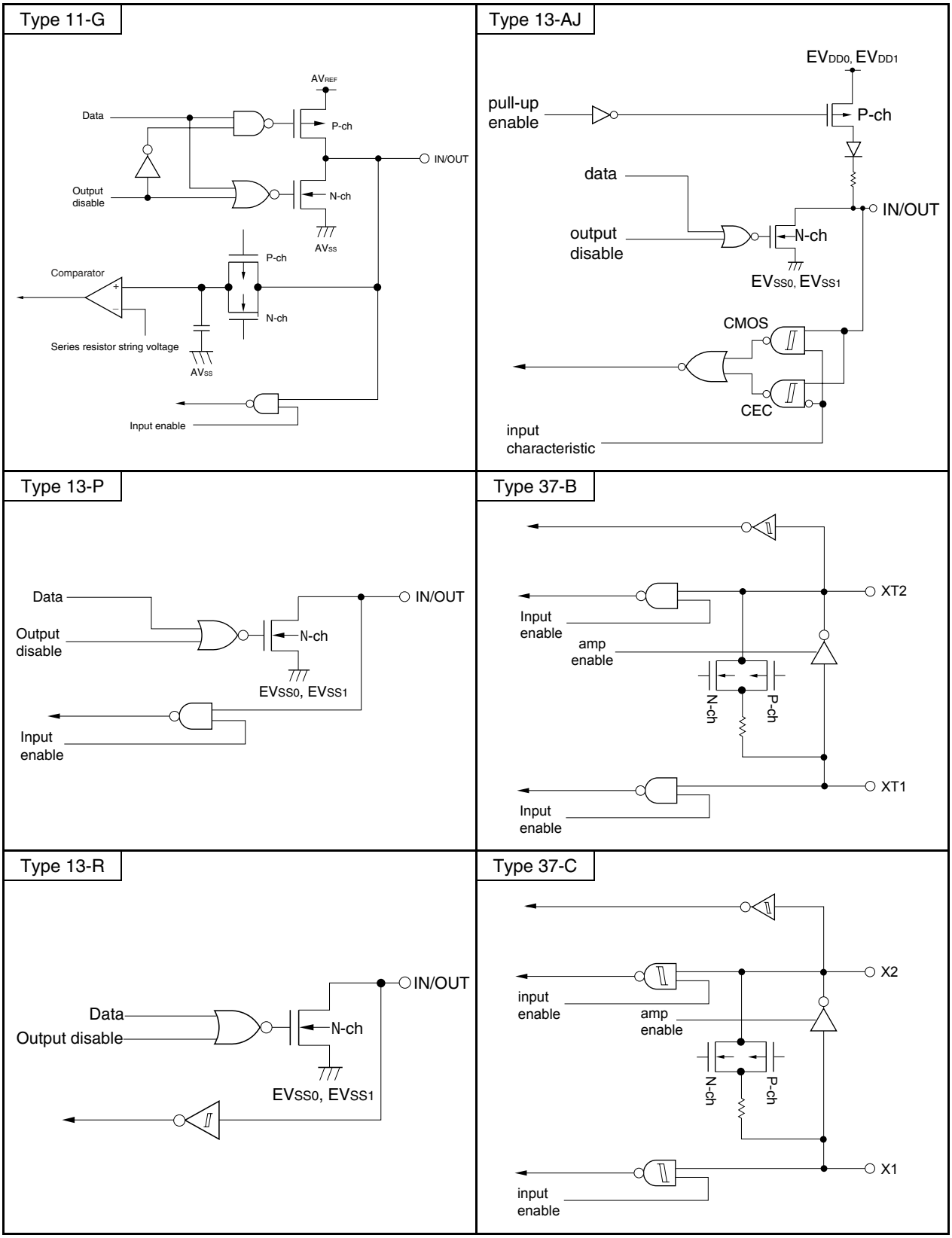
Note P150/ANI8 to P157/ANI15 are set in the digital input port mode after release of reset.

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



Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

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



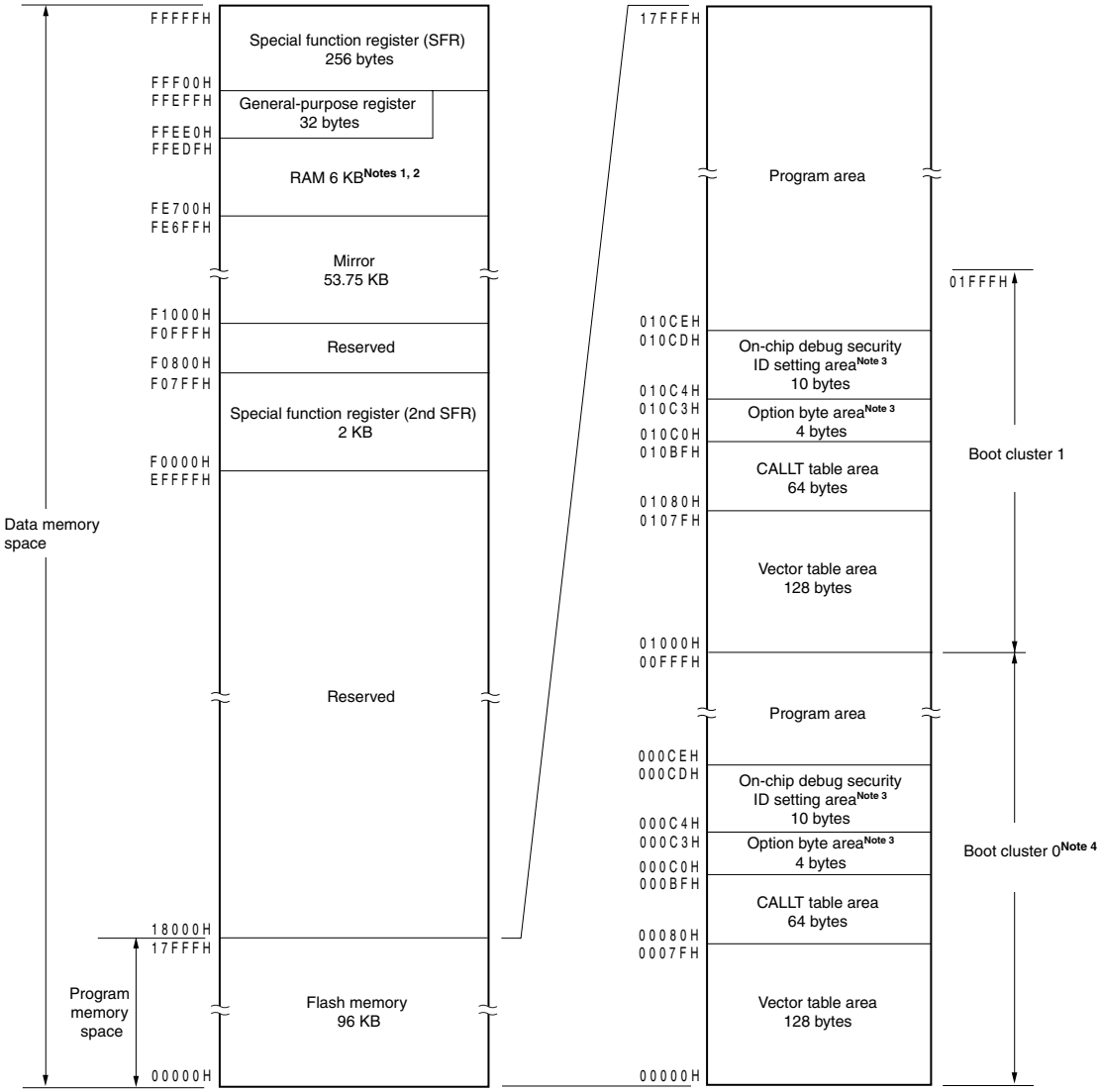
Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the 78K0R/Kx3-C can access a 1 MB memory space. Figures 3-1 and 3-2 show the memory maps.

Figure 3-1. Memory Map (μPD78F1846A, 78F1848A)



Notes 1. While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory.

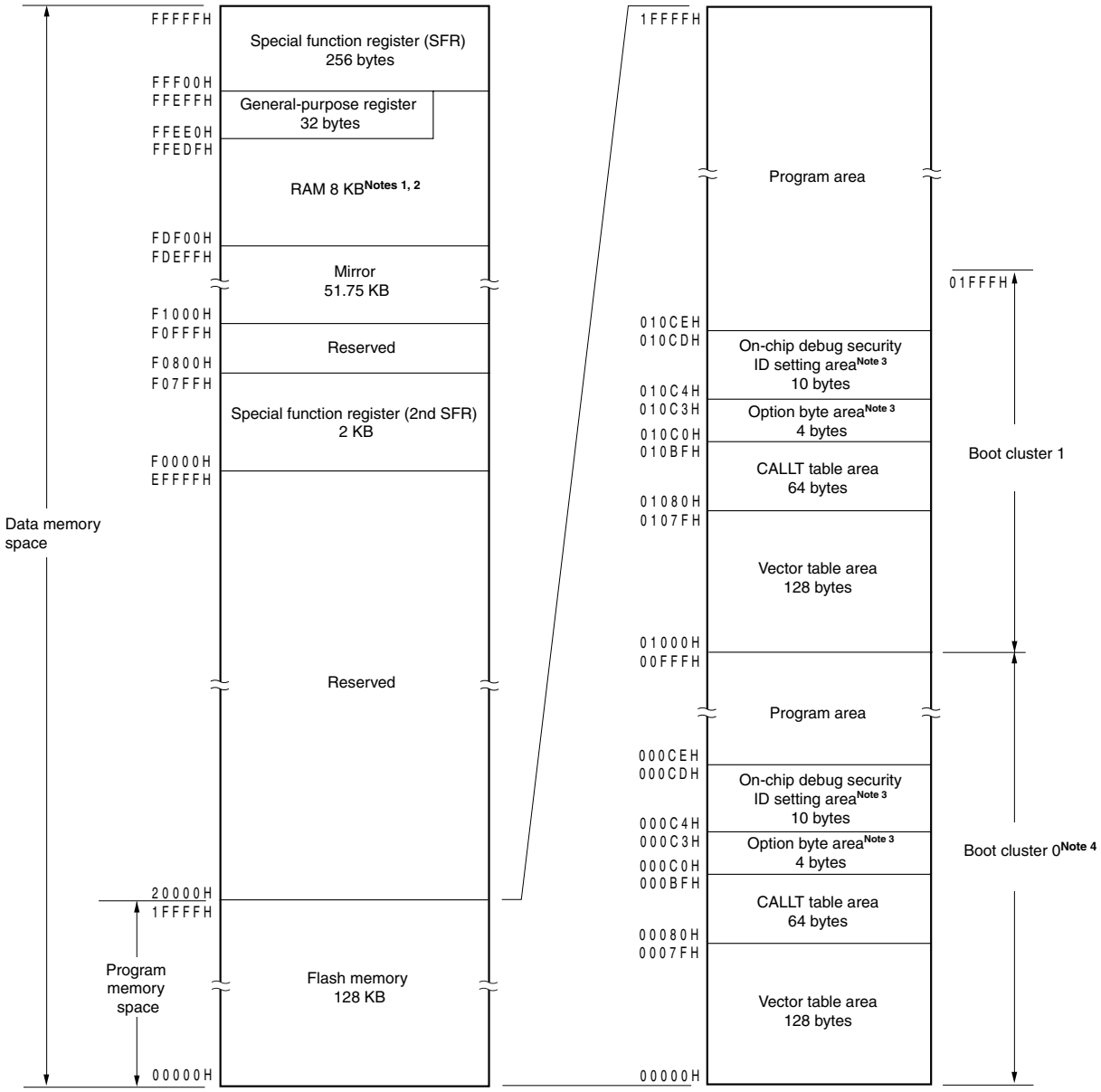
2. Instructions can be executed from the RAM area excluding the general-purpose register area.

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

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

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **25.7 Security Setting**).

Figure 3-2. Memory Map (μPD78F1847A, 78F1849A)



Notes 1. While using the self-programming function, the area of FFE20H to FFEFFH and FDF00H to FE2FFH cannot be used as a stack memory.

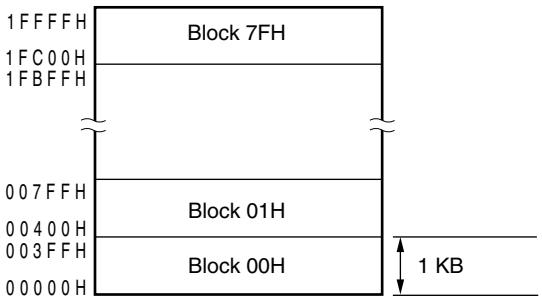
2. Instructions can be executed from the RAM area excluding the general-purpose register area.

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

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

4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **25.7 Security Setting**).

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



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

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

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

Remark μPD78F1846A, 78F1848A: Block numbers 00H to 5FH
μPD78F1847A, 78F1849A: Block numbers 00H to 7FH

3.1.1 Internal program memory space

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

78K0R/Kx3-C products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F1846A, 78F1848A	Flash memory	98304 × 8 bits (00000H to 17FFFH)
μPD78F1847A, 78F1849A		131072 × 8 bits (00000H to 1FFFFH)

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	$\overline{\text{RESET}}$ input, POC, LVI, WDT, TRAP	00034H	INTAD
		00036H	INTRTC
00004H	INTWDTI	00038H	INTRTCI
00006H	INTLVI	0003AH	INTKR
00008H	INTP0	0003CH	INTST2/INTCSI20/INTIIC20
0000AH	INTP1	0003EH	INTRERR3
0000CH	INTP2	00040H	INTRA3
0000EH	INTP3	00042H	INTTM04
00010H	INTP4	00044H	INTTM05
00012H	INTP5	00046H	INTTM06
00014H	INTDA	00048H	INTTM07
00016H	INTCE	0004AH	INTSR2
00018H	INTERR	0004CH	INTP7
0001AH	INTDMA0	0004EH	INTP8
0001CH	INTDMA1	00050H	INTRERR2
0001EH	INTST0/INTCSI00	00052H	INTRA2
00020H	INTSR0/INTCSI01	00054H	INTRERR1
00022H	INTSRE0	00056H	INTTM10
00024H	INTST1/INTCSI10/INTIIC10	00058H	INTTM11
00026H	INTSR1	0005AH	INTTM12
00028H	INTSRE1	0005CH	INTSRE2
0002AH	INTIICA	0005EH	INTRA1
0002CH	INTTM00	00060H	INTRERR0
0002EH	INTTM01	00062H	INTRA0
00030H	INTTM02	0007EH	BRK
00032H	INTTM03		

(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/Kx3-C mirror 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)).

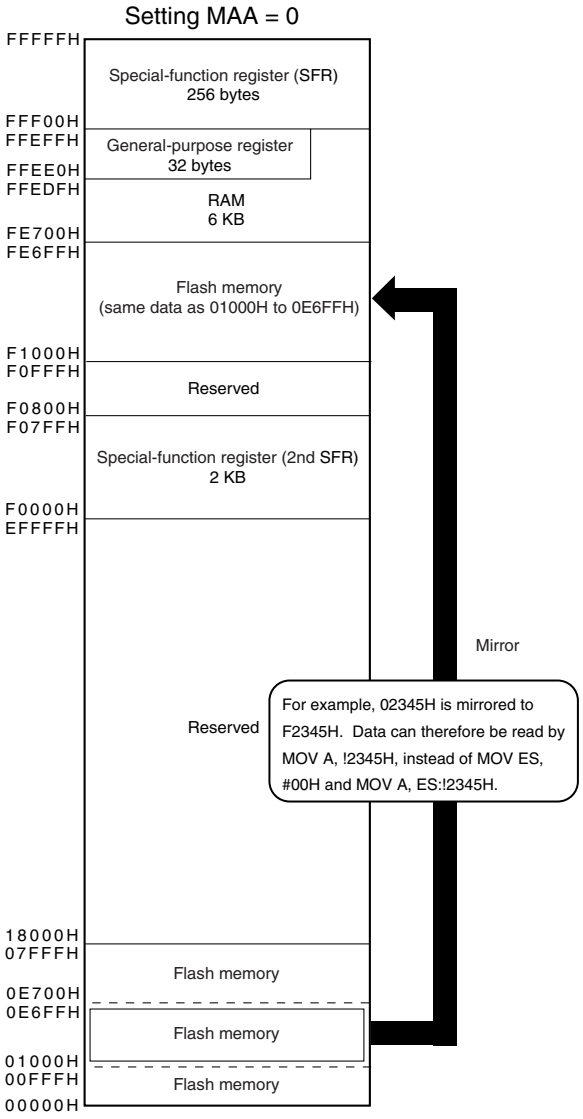
By reading data from F0000H to FFFFFH, an instruction that does not have the ES registers as an operand can be used, and thus the contents of the data flash can be read with the shorter code. However, the data flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

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

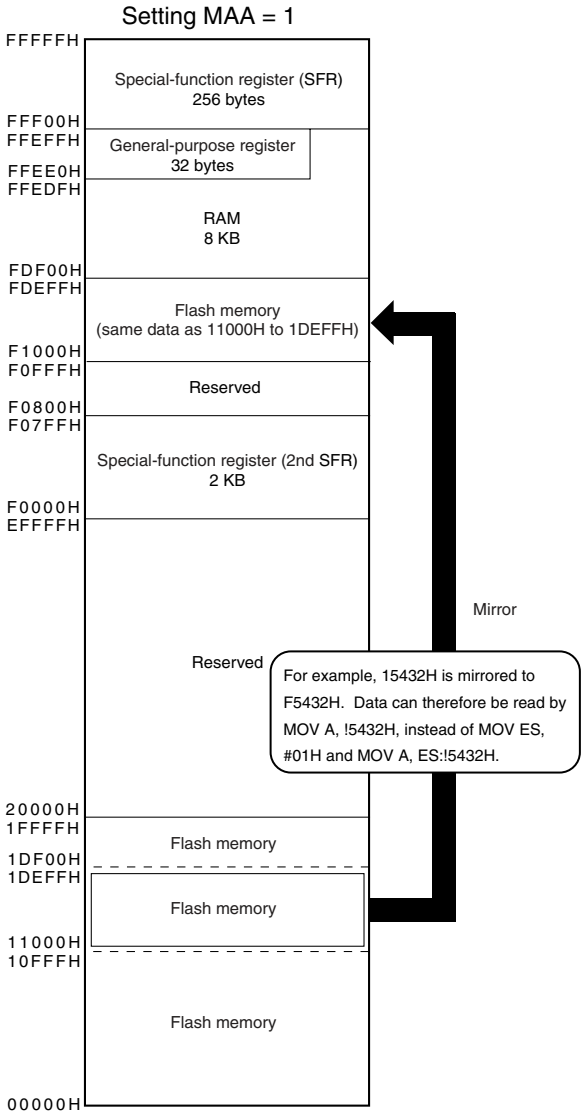
The mirror area can only be read and no instruction can be fetched from this area.

The following show examples.

Example 1 μPD78F1846A, 78F1848A
(Flash memory: 96 KB, RAM: 6 KB)



Example 2 μPD78F1847A, 78F1849A
(Flash memory: 128 KB, RAM: 8 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 3-3. Format 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/Kx3-C products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
μPD78F1846A, 78F1848A	6144 × 8 bits (FE700H to FFEFFH)
μPD78F1847A, 78F1849A	8192 × 8 bits (FDF00H 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 FFE20H to FFEFFH cannot be used as stack memory. Furthermore, the area FDF00H to FE2FFH also cannot be used as stack memory with the μPD78F1847A and 78F1849A.

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.

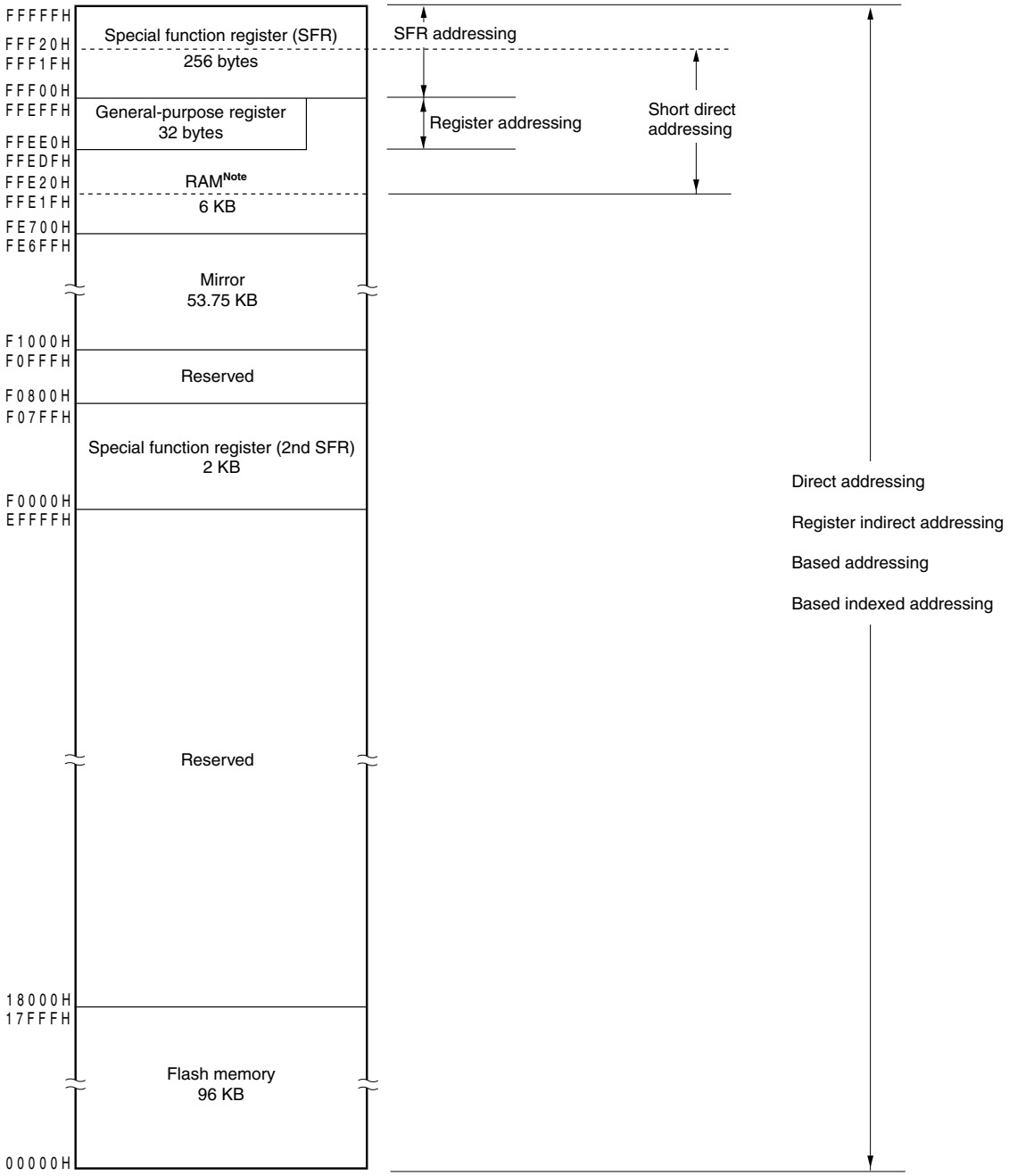
Caution Do not access addresses to which the 2nd SFR is not assigned.

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.

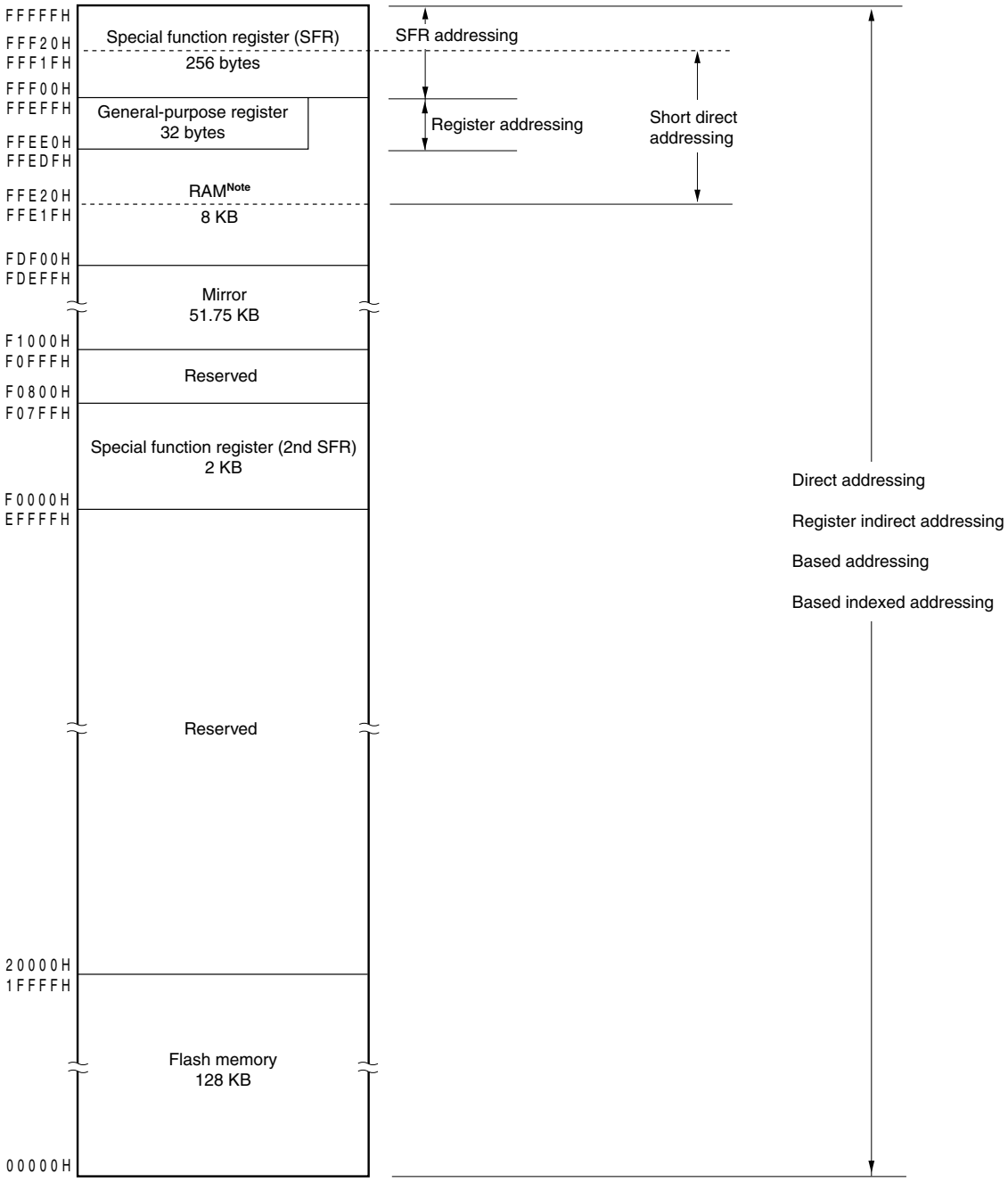
Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the 78K0R/Kx3-C, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Figures 3-4 and 3-5 show correspondence between data memory and addressing.

Figure 3-4. Correspondence Between Data Memory and Addressing (μPD78F1846A, 78F1848A)



Note While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory.

Figure 3-5. Correspondence Between Data Memory and Addressing (μPD78F1847A, 78F1849A)



Note While using the self-programming function, the area of FFE20H to FFEFFH and FDF00H to FE2FFH cannot be used as a stack memory.

3.2 Processor Registers

The 78K0R/Kx3-C 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)

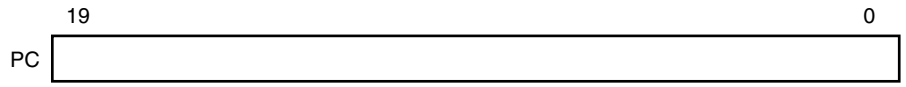
The program counter is a 20-bit register that holds the address information of the next program to be executed.

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 3-6. Format of Program Counter



(2) Program status word (PSW)

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

Program status word contents are stored in the stack area upon 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 3-7. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

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

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

When 1, the IE flag is set to the interrupt enabled (EI) state and 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)**) cannot be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-8. Format of Stack Pointer

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

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 3-9.

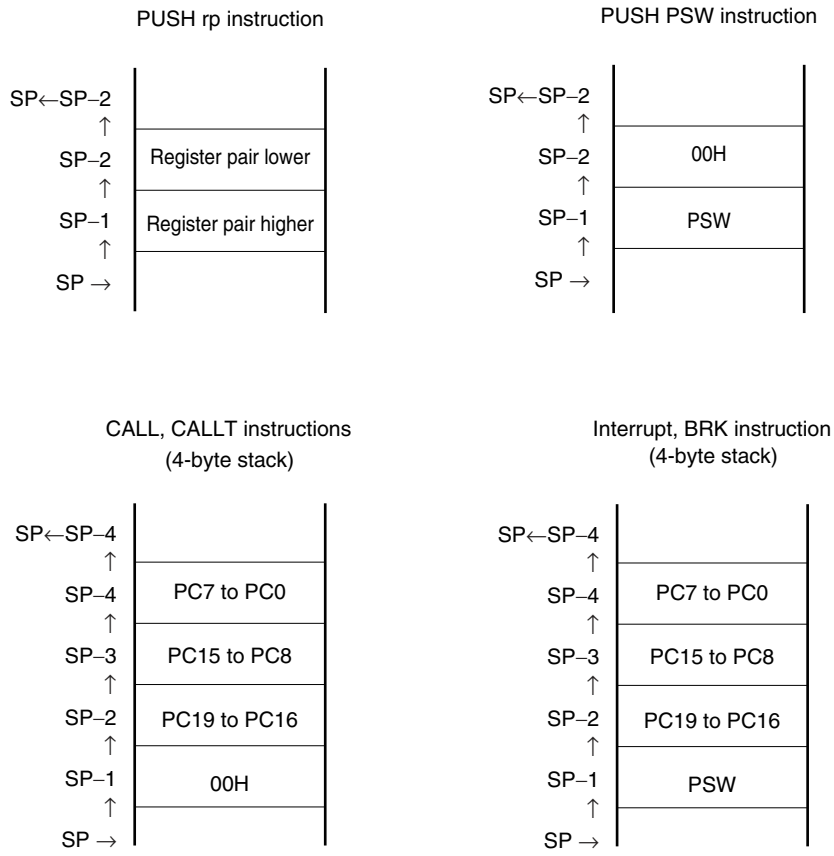
Cautions 1. Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

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 FFE20H to FFEFFH cannot be used as stack memory. Furthermore, the area FDF00H to FE2FFH also cannot be used as stack memory with the μPD78F1847A and 78F1849A.

Figure 3-9. 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 consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

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

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

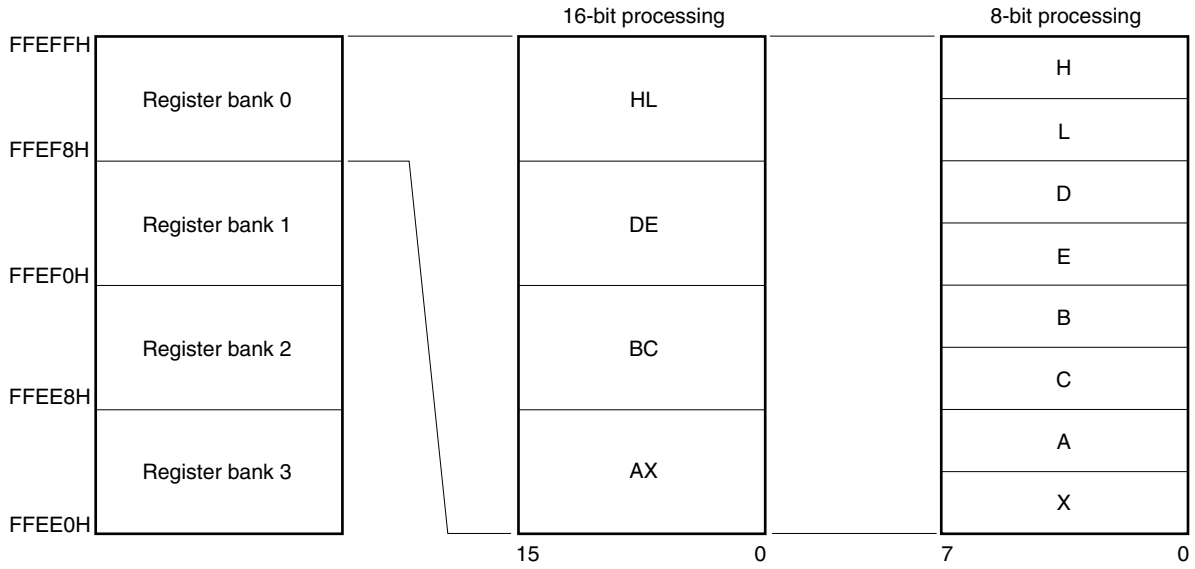
Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

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

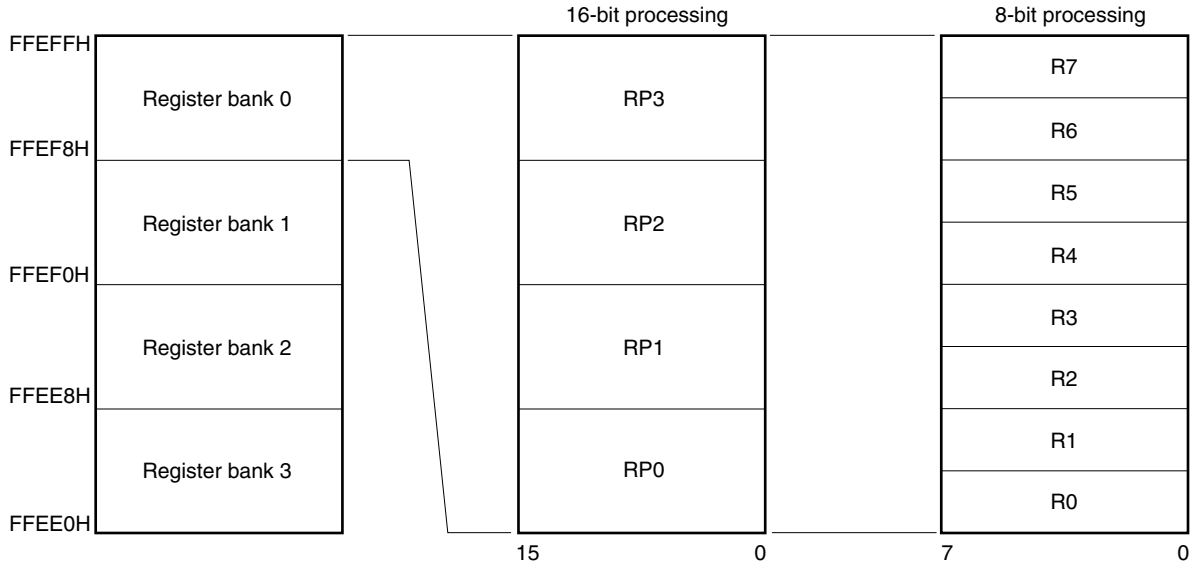
2. While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory. Furthermore, the areas of FDF00H to FE2FFH also cannot be used with the *μ*PD78F1847A and 78F1849A, respectively.

Figure 3-10. Configuration of General-Purpose Registers

(a) Function name



(b) Absolute name



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 3-11. Configuration of ES and CS Registers

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

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

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[Note]	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		–		–	–		
FFF18H	Timer data register 00	TDR00	R/W	–	–	√	0000H
FFF19H							
FFF1AH	Timer data register 01	TDR01	R/W	–	–	√	0000H
FFF1BH							
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[Note]	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
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

Note 78K0R/KG3-C only

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	
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
FFF3EH	Timer input select register 0	TIS0		R/W	√	√	–	00H
FFF3FH	Timer input select register 1	TIS1		R/W	√	√	–	00H
FFF40H	Remote controller receive data register 0	RMDR0		R	–	√	–	00H
FFF41H	Remote controller receive counter register 0	RMSCR0		R	–	√	–	00H
FFF42H	Remote controller receive shift register 0	RMSR0		R	–	√	–	00H
FFF43H	Remote control reception error bit detection register 0	RMERBD0		R	–	√	–	00H
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	–	√	√	0000H
FFF45H		–			–	–		
FFF46H	Serial data register 03	RXD1	SDR03	R/W	–	√	√	0000H
FFF47H		–			–	–		
FFF48H	Serial data register 10	TXD2/ SIO20	SDR10	R/W	–	√	√	0000H
FFF49H		–			–	–		
FFF4AH	Serial data register 11	RXD2	SDR11	R/W	–	√	√	0000H
FFF4BH		–			–	–		
FFF4CH	Remote control reception error bit detection shift register 0	RMERBSR0		R	–	√	–	00H
FFF4DH	Remote controller receive interrupt status register 0	RMINTS0		R/W	–	√	–	00H
FFF4EH	Remote controller receive data register 1	RMDR1		R/W	–	√	–	00H
FFF4FH	Remote controller receive counter register 1	RMSCR1		R/W	–	√	–	00H
FFF50H	IICA shift register	IICA		R/W	–	√	–	00H
FFF51H	IICA status register	IICS		R	√	√	–	00H
FFF52H	IICA flag register	IICF		R/W	√	√	–	00H
FFF54H	Remote controller receive shift register 1	RMSR1		R/W	–	√	–	00H
FFF55H	Remote control reception error bit detection register 1	RMERBD1		R/W	–	√	–	00H
FFF56H	Remote control reception error bit detection shift register 1	RMERBSR1		R/W	–	√	–	00H
FFF57H	Remote controller receive interrupt status register 1	RMINTS1		R/W	–	√	–	00H
FFF58H	Remote controller receive data register 2	RMDR2		R/W	–	√	–	00H
FFF59H	Remote controller receive counter register 2	RMSCR2		R/W	–	√	–	00H
FFF5AH	Remote controller receive shift register 2	RMSR2		R/W	–	√	–	00H
FFF5BH	Remote control reception error bit detection register 2	RMERBD2		R/W	–	√	–	00H
FFF5CH	Remote control reception error bit detection shift register 2	RMERBSR2		R/W	–	√	–	00H
FFF5DH	Remote controller receive interrupt status register 2	RMINTS2		R/W	–	√	–	00H

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	
FFF5EH	Remote controller receive data register 3	RMDR3	R/W	–	√	–	00H
FFF5FH	Remote controller receive counter register 3	RMSCR3	R/W	–	√	–	00H
FFF60H	Remote controller receive shift register 3	RMSR3	R/W	–	√	–	00H
FFF61H	Remote control reception error bit detection register 3	RMERBD3	R/W	–	√	–	00H
FFF62H	Remote control reception error bit detection shift register 3	RMERBSR3	R/W	–	√	–	00H
FFF63H	Remote controller receive interrupt status register 3	RMINTS3	R/W	–	√	–	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							
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							
FFF78H	CEC transmission buffer register	CTXD	R/W	–	√	–	
FFF79H	CEC reception buffer register	CRXD	R	–	√	–	
FFF7AH	CEC communication error status register	CECES	R	–	√	–	
FFF7BH	CEC communication status register	CECS	R	–	√	–	
FFF7CH	CEC communication error flag clear trigger register	CECFC	R/W	√	√	–	
FFF7DH	CEC control register 0	CECCTL0	R/W	√	√	–	
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[Note]
FFF95H	Week count register	WEEK	R/W	–	√	–	00H
FFF96H	Day count register	DAY	R/W	–	√	–	01H
FFF97H	Month count register	MONTH	R/W	–	√	–	01H
FFF98H	Year count register	YEAR	R/W	–	√	–	00H
FFF99H	Watch error correction register	SUBCUD	R/W	–	√	–	00H

Note The value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

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	
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[Note 1]
FFFA9H	Low-voltage detection register	LVIM		R/W	√	√	–	00H[Note 2]
FFFAAH	Low-voltage detection level select register	LVIS		R/W	√	√	–	0EH[Note 3]
FFFABH	Watchdog timer enable register	WDTE		R/W	–	√	–	1A/9A[Note 4]
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
FFFBAH	DMA mode control register 0	DMC0		R/W	√	√	–	00H
FFFBBH	DMA mode control register 1	DMC1		R/W	√	√	–	00H
FFFBCH	DMA operation control register 0	DRC0		R/W	√	√	–	00H
FFFBDH	DMA operation control register 1	DRC1		R/W	√	√	–	00H
FFFBEH	Back ground event control register	BECTL		R/W	√	√	–	00H
FFFC0H	–	PFCMD[Note 5]		–	–	–	–	Undefined
FFFC2H	–	PFS[Note 5]		–	–	–	–	Undefined
FFFC4H	–	FLPMC[Note 5]		–	–	–	–	Undefined

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

2. The reset value of LVIM varies depending on the reset source and the setting of the option byte.

3. The reset value of LVIS varies depending on the reset source.

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

5. 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	
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
FFFDDH	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
FFFEAH	Priority specification flag register 01L	PR01L	PR01	R/W	√	√	√	FFH
FFFEBH	Priority specification flag register 01H	PR01H		R/W	√	√		FFH
FFFECH	Priority specification flag register 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFEDH	Priority specification flag register 10H	PR10H		R/W	√	√		FFH
FFFEEH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	Priority specification flag register 11H	PR11H		R/W	√	√		FFH
FFFF0H	Multiplication/division data register A (L)	MDAL/MULA		R/W	–	–	√	0000H
FFFF1H								
FFFF2H	Multiplication/division data register A (H)	MDAH/MULB		R/W	–	–	√	0000H
FFFF3H								
FFFF4H	Multiplication/division data register B (H)	MDBH/MULOH		R/W	–	–	√	0000H
FFFF5H								
FFFF6H	Multiplication/division data register B (L)	MDBL/MULOL		R/W	–	–	√	0000H
FFFF7H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	–	00H

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

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

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

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

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

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe 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
  "√" 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 the 2nd SFR is 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 (1/7)

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 8Note	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 13Note	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
F0041H	Port input mode register 1	PIM1	R/W	√	√	–	00H
F0046H	Port input mode register 6	PIM6	R/W	√	√	–	00H
F004EH	Port input mode register 14	PIM14	R/W	√	√	–	00H
F0050H	Port output mode register 0	POM0	R/W	√	√	–	00H
F0051H	Port output mode register 1	POM1	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
F0076H	Port function register 6	PF6	R/W	√	√	–	00H
F007BH	Port function register 11	PF11	R/W	√	√	–	00H
F00E0H	Multiplication/division data register C (L)	MDCL	R	–	–	√	0000H
F00E2H	Multiplication/division data register C (H)	MDCH	R	–	–	√	0000H
F00E8H	Multiplication/division control register	MDUC	R/W	√	√	–	00H
F00F0H	Peripheral enable register 0	PER0	R/W	√	√	–	00H
F00F1H	Peripheral enable register 1	PER1	R/W	√	√	–	00H
F00F3H	Operation speed mode control register	OSMC	R/W	–	√	–	00H
F00F4H	Regulator mode control register	RMC	R/W	–	√	–	00H
F00F6H	20 MHz internal high-speed oscillation control register	DSCCTL	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		–			–	–		

Note 78K0R/KG3-C only

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	–	√	√	0000H
F0109H		–			–	–		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	–	√	√	0000H
F010BH		–			–	–		
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	–	√	√	0000H
F010DH		–			–	–		
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	–	√	√	0000H
F010FH		–			–	–		
F0110H	Serial mode register 00	SMR00		R/W	–	–	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	–	–	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	–	–	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	–	–	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	–	–	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	–	–	√	0087H
F011FH								
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		–			–	–		
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	–	√	√	0000H
F0149H		–			–	–		

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	–	√	√	0000H
F014BH		–			–	–		
F0150H	Serial mode register 10	SMR10		R/W	–	–	√	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	–	–	√	0020H
F0153H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	–	–	√	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	–	–	√	0087H
F015BH								
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								
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								

Table 3-6. Extended SFR (2nd SFR) List (4/7)

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

Table 3-6. Extended SFR (2nd SFR) List (5/7)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01C0H	Timer counter register 10	TCR10		R	–	–	√	FFFFH
F01C1H								
F01C2H	Timer counter register 11	TCR11		R	–	–	√	FFFFH
F01C3H								
F01C4H	Timer counter register 12	TCR12		R	–	–	√	FFFFH
F01C5H								
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								
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		–			–	–		
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		–			–	–		
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
F01B7H		–			–	–		
F0230H	IICA control register 0	IICCTL0		R/W	√	√	–	00H
F0231H	IICA control register 1	IICCTL1		R/W	√	√	–	00H
F0232H	IICA low-level width setting register	IICWL		R/W	–	√	–	FFH
F0233H	IICA high-level width setting register	IICWH		R/W	–	√	–	FFH
F0234H	Slave address register	SVA		R/W	–	√	–	00H
F0300H	CEC local address setting register	CADR		R/W	–	–	√	0000H
F0302H	CEC control register 1	CECCTL1		R/W	√	√	–	00H
F0304H	CEC transmission start bit width setting register	STATB		R/W	–	–	√	0000H
F0306H	CEC transmission start bit low width setting register	STATL		R/W	–	–	√	0000H
F0308H	CEC transmission logical 0 low width setting register	LGC0L		R/W	–	–	√	0000H
F030AH	CEC transmission logical 1 low width setting register	LGC1L		R/W	–	–	√	0000H

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F030CH	CEC transmission data bit width setting register	DATB	R/W	–	–	√	0000H
F030EH	CEC reception data sampling time setting register	NOMT	R/W	–	–	√	0000H
F0310H	CEC reception start bit minimum low width setting register	STATLL	R/W	–	–	√	0000H
F0312H	CEC reception start bit maximum low width setting register	STATLH	R/W	–	–	√	0000H
F0314H	CEC reception start bit minimum bit width setting register	STATBL	R/W	–	–	√	0000H
F0316H	CEC reception start bit maximum bit width setting register	STATBH	R/W	–	–	√	0000H
F0318H	CEC reception logical 0 minimum low width setting register	LGC0LL	R/W	–	–	√	0000H
F031AH	CEC reception logical 0 maximum low width setting register	LGC0LH	R/W	–	–	√	0000H
F031CH	CEC reception logical 1 minimum low width setting register	LGC1LL	R/W	–	–	√	0000H
F031EH	CEC reception logical 1 maximum low width setting register	LGC1LH	R/W	–	–	√	0000H
F0320H	CEC reception data bit minimum bit width setting register	DATBL	R/W	–	–	√	0000H
F0322H	CEC reception data bit maximum bit width setting register	DATBH	R/W	–	–	√	0000H
F0324H	CEC data bit reference width setting register	NOMP	R/W	–	–	√	0000H
F0330H	Remote controller receive control 1 register 02	RMCN102	R/W	√	√	–	00H
F0331H	Remote controller receive control 2 register 02	RMCN202	R/W	√	√	–	00H
F0332H	Remote controller receive GPBS compare register 02	RMGPBS02	R/W	–	–	√	0000H
	Remote controller receive GPLS compare register 02	RMGPLS02	R/W	–	√	–	00H
F0333H	Remote controller receive GPLL compare register 02	RMGPLL02	R/W	–	√	–	00H
F0334H	Remote controller receive GPBL compare register 02	RMGPBL02	R/W	–	–	√	0000H
	Remote controller receive GPHS compare register 02	RMGPHS02	R/W	–	√	–	00H
F0335H	Remote controller receive GPHL compare register 02	RMGPHL02	R/W	–	√	–	00H
F0336H	Remote controller receive DB0S compare register 02	RMDB0S02	R/W	–	–	√	0000H
	Remote controller receive DLS compare register 02	RMDLS02	R/W	–	√	–	00H
F0337H	Remote controller receive DLL compare register 02	RMDLL02	R/W	–	√	–	00H
F0338H	Remote controller receive DB0L compare register 02	RMDB0L02	R/W	–	–	√	0000H
	Remote controller receive DH0S compare register 02	RMDH0S02	R/W	–	√	–	00H
F0339H	Remote controller receive DH0L compare register 02	RMDH0L02	R/W	–	√	–	00H

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F033AH	Remote controller receive DB1S compare register 02	RMDB1S02	R/W	–	–	√	0000H
	Remote controller receive DH1S compare register 02	RMDH1S02	R/W	–	√	–	00H
F033BH	Remote controller receive DH1L compare register 02	RMDH1L02	R/W	–	√	–	00H
F033CH	Remote controller receive DB1L compare register 02	RMDB1L02	R/W	–	–	√	0000H
F033EH	Remote controller receive end width select register L02	RMERL02	R/W	–	√	–	00H
F033FH	Remote controller receive end width select register H02	RMERH02	R/W	–	√	–	00H
F0340H	Remote controller receive noise elimination period setting register 02	RMNCP102	R/W	–	√	–	00H
F0342H	Remote controller receive control 1 register 13	RMCN113	R/W	√	√	–	00H
F0343H	Remote controller receive control 2 register 13	RMCN213	R/W	√	√	–	00H
F0344H	Remote controller receive GPBS compare register 13	RMGPBS13	R/W	–	–	√	0000H
	Remote controller receive GPLS compare register 13	RMGPLS13	R/W	–	√	–	00H
F0345H	Remote controller receive GPLL compare register 13	RMGPLL13	R/W	–	√	–	00H
F0346H	Remote controller receive GPBL compare register 13	RMGPBL13	R/W	–	–	√	0000H
	Remote controller receive GPHS compare register 13	RMGPHS13	R/W	–	√	–	00H
F0347H	Remote controller receive GPHL compare register 13	RMGPHL13	R/W	–	√	–	00H
F0348H	Remote controller receive DB0S compare register 13	RMDB0S13	R/W	–	–	√	0000H
	Remote controller receive DLS compare register 13	RMDLS13	R/W	–	√	–	00H
F0349H	Remote controller receive DLL compare register 13	RMDLL13	R/W	–	√	–	00H
F034AH	Remote controller receive DB0L compare register 13	RMDB0L13	R/W	–	–	√	0000H
	Remote controller receive DH0S compare register 13	RMDH0S13	R/W	–	√	–	00H
F034BH	Remote controller receive DH0L compare register 13	RMDH0L13	R/W	–	√	–	00H
F034CH	Remote controller receive DB1S compare register 13	RMDB1S13	R/W	–	–	√	0000H
	Remote controller receive DH1S compare register 13	RMDH1S13	R/W	–	√	–	00H
F034DH	Remote controller receive DH1L compare register 13	RMDH1L13	R/W	–	√	–	00H
F034EH	Remote controller receive DB1L compare register 13	RMDB1L13	R/W	–	–	√	0000H
F0350H	Remote controller receive end width select register L13	RMERL13	R/W	–	√	–	00H
F0351H	Remote controller receive end width select register H13	RMERH13	R/W	–	√	–	00H
F0352H	Remote controller receive noise elimination period setting register 13	RMNCP113	R/W	–	√	–	00H
F0353H	Remote controller receive data slew control register	RMSW	R/W	–	√	–	00H

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

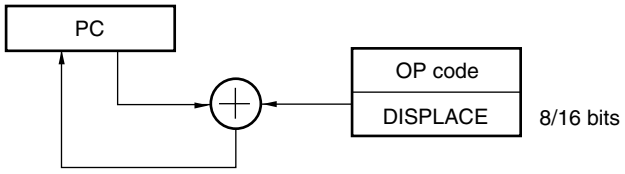
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

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

Figure 3-12. Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

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

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

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

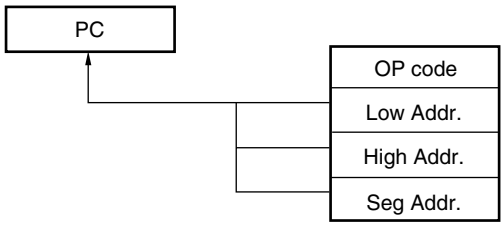
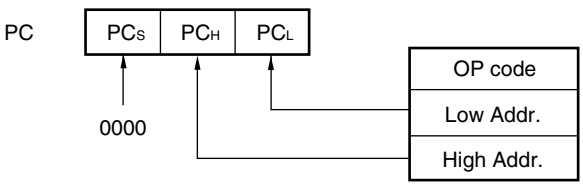


Figure 3-14. Example of CALL !addr16/BR !addr16



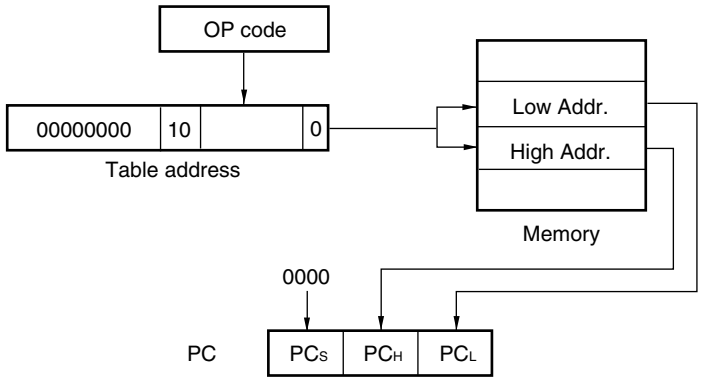
3.3.3 Table indirect addressing

[Function]

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

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

Figure 3-15. Outline of Table Indirect Addressing

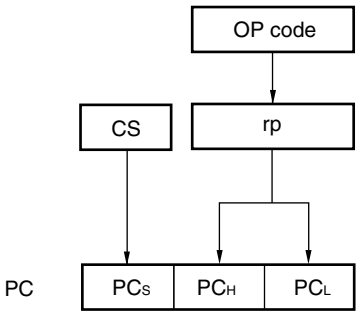


3.3.4 Register direct addressing

[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 3-16. Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

[Function]

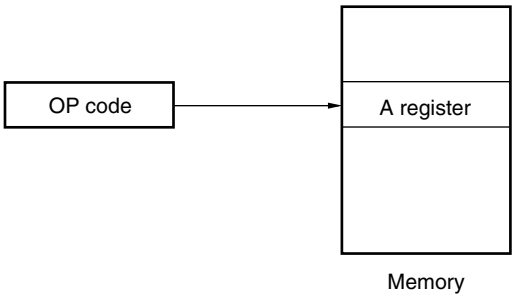
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[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 3-17. Outline of Implied Addressing



3.4.2 Register addressing

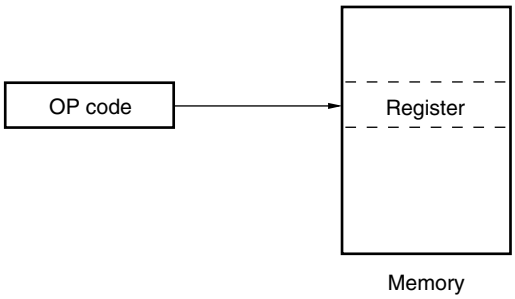
[Function]

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

[Operand format]

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

Figure 3-18. Outline of Register Addressing



3.4.3 Direct addressing

[Function]

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

[Operand format]

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

Figure 3-19. Example of ADDR16

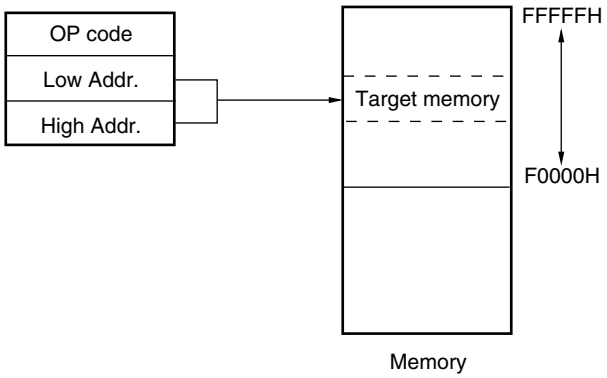
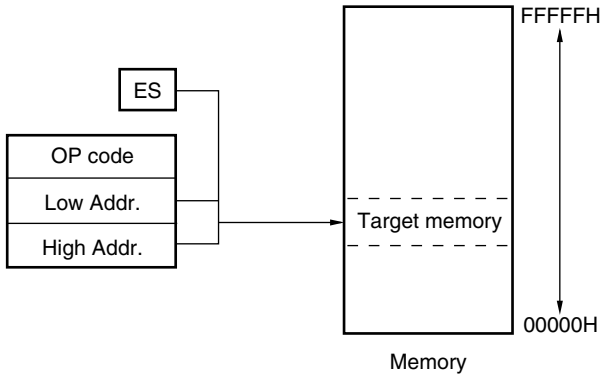


Figure 3-20. Example of ES:ADDR16



3.4.4 Short direct addressing

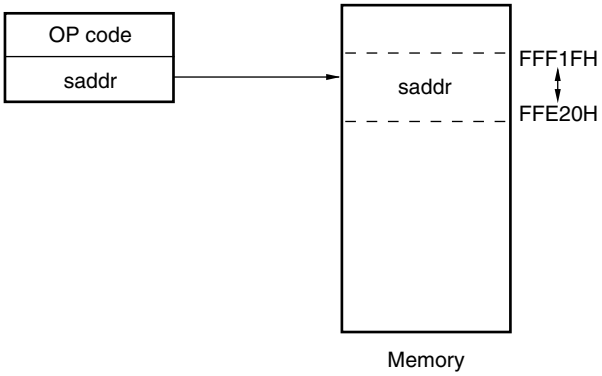
[Function]

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 3-21. Outline of Short Direct Addressing



Remark SADDR and SADDRP are used to describe the values of addresses FE20H to FF1FH with 16-bit immediate data (higher 4 bits of actual address are omitted), and the values of addresses FFE20H to FFF1FH with 20-bit immediate data.

Regardless of whether 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

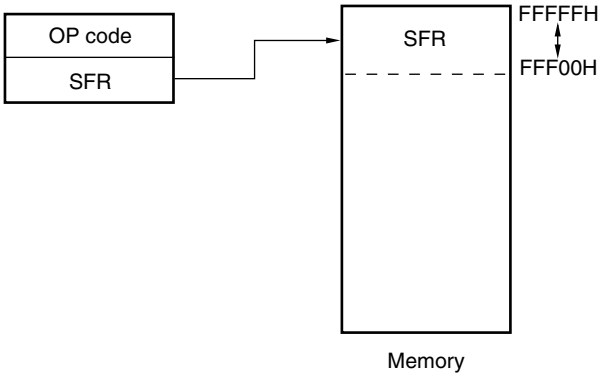
[Function]

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-manipulable SFR name (even address only)

Figure 3-22. Outline of SFR Addressing



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

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

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

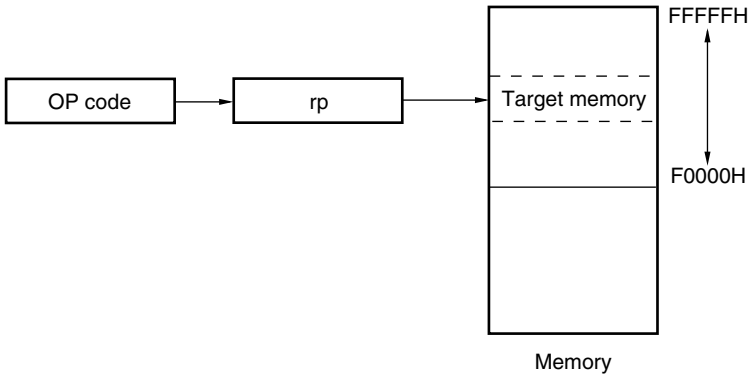
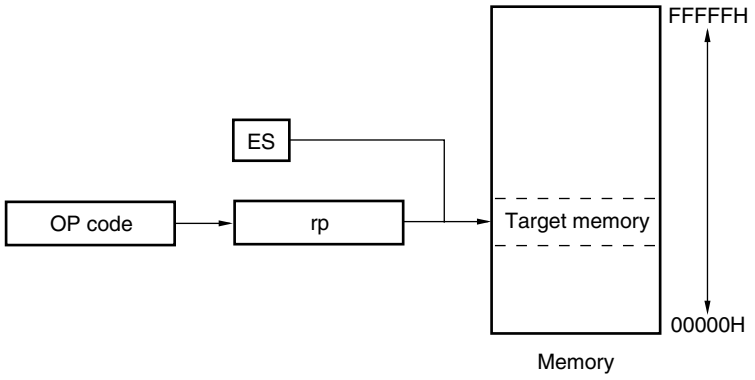


Figure 3-24. Example of ES:[DE], ES:[HL]



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 3-25. Example of [SP+byte]

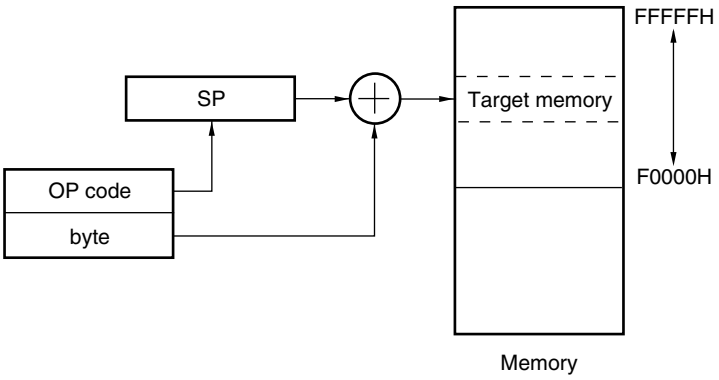


Figure 3-26. Example of [HL + byte], [DE + byte]

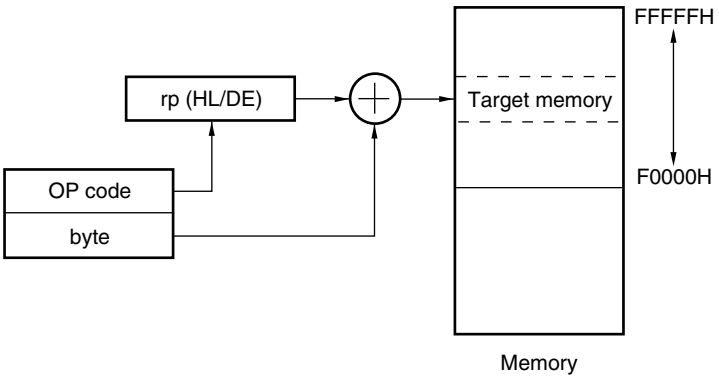


Figure 3-27. Example of word[B], word[C]

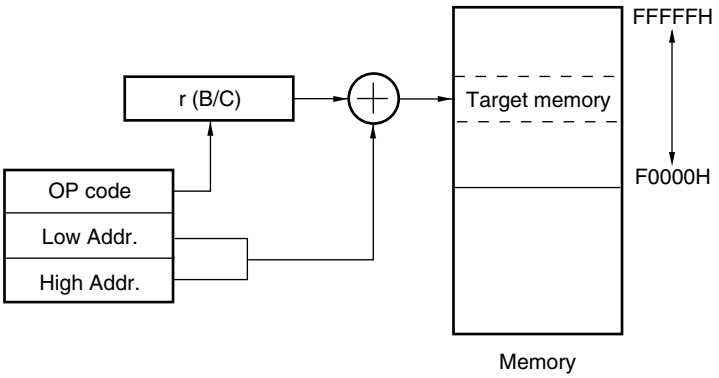


Figure 3-28. Example of word[BC]

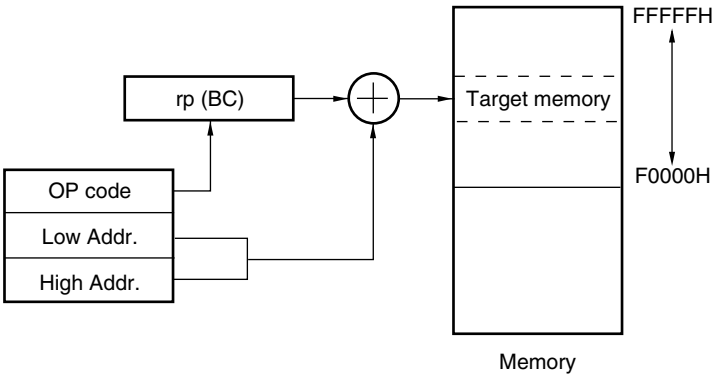


Figure 3-29. Example of ES:[HL + byte], ES:[DE + byte]

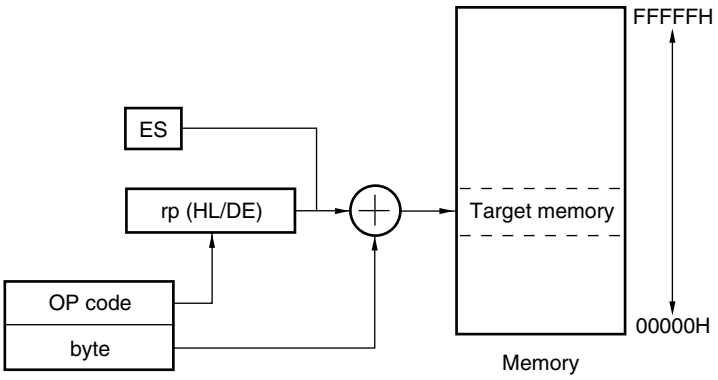


Figure 3-30. Example of ES:word[B], ES:word[C]

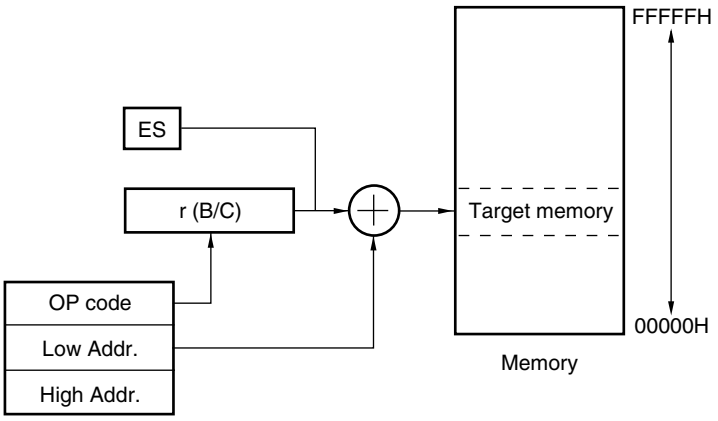
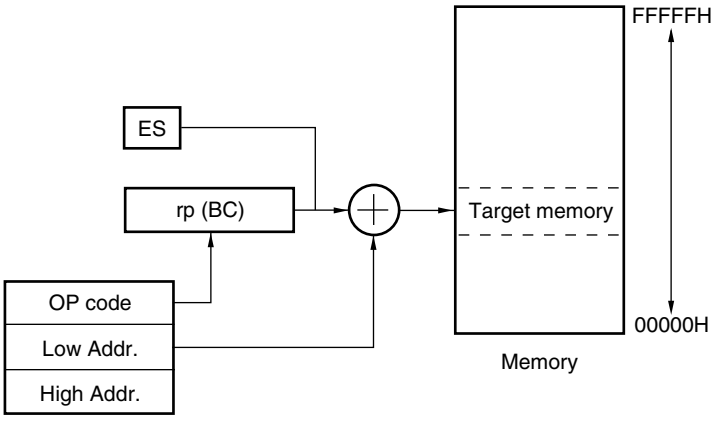


Figure 3-31. Example of ES:word[BC]



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

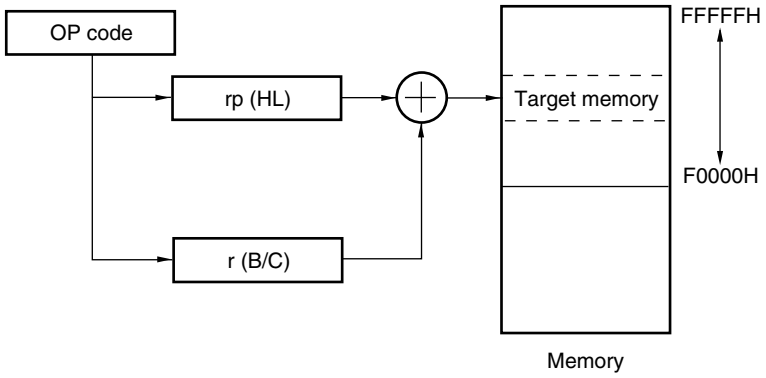
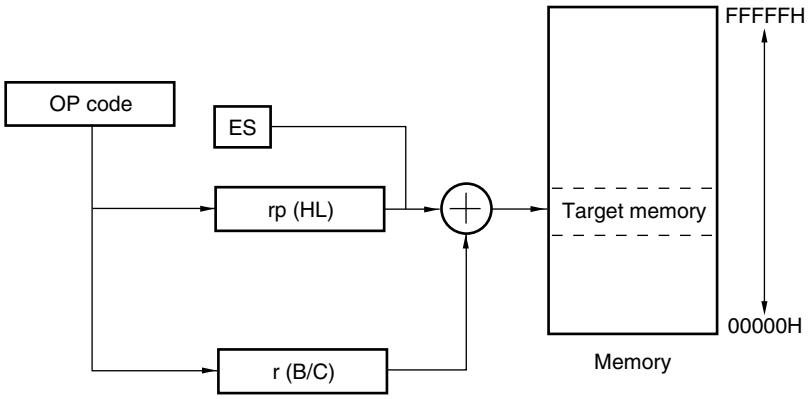


Figure 3-33. Example of ES:[HL+B], ES:[HL+C]



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

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

Table 4-1. Pin I/O Buffer Power Supplies (AV_{REF}, EV_{DD}, V_{DD})

- 78K0R/KF3-C: 80-pin plastic LQFP (fine pitch) (12x12)

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27, P150 to P153
EV_{DD}	• Port pins other than P20 to P27, P121 to P124, and P150 to P153 • $\overline{RESET}$ and FLMD0 pins
V_{DD}	• P121 to P124 • Pins other than port pins (excluding $\overline{RESET}$ and FLMD0 pins)

Table 4-2. Pin I/O Buffer Power Supplies (AV_{REF}, EV_{DD0}, EV_{DD1}, V_{DD})

- 78K0R/KG3-C: 100-pin plastic LQFP (fine pitch) (14x14)

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27, P150 to P157
EV_{DD0}, EV_{DD1}	• Port pins other than P20 to P27, P121 to P124, and P150 to P157 • $\overline{RESET}$ and FLMD0 pins
V_{DD}	• P121 to P124 • Pins other than port pins (excluding $\overline{RESET}$ and FLMD0 pins)

78K0R/Kx3-C products are provided with the digital I/O ports shown in Figures 4-1 and 4-2, which enable variety of control operations. The functions of each port are shown in Table 4-3.

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 (78K0R/KF3-C)

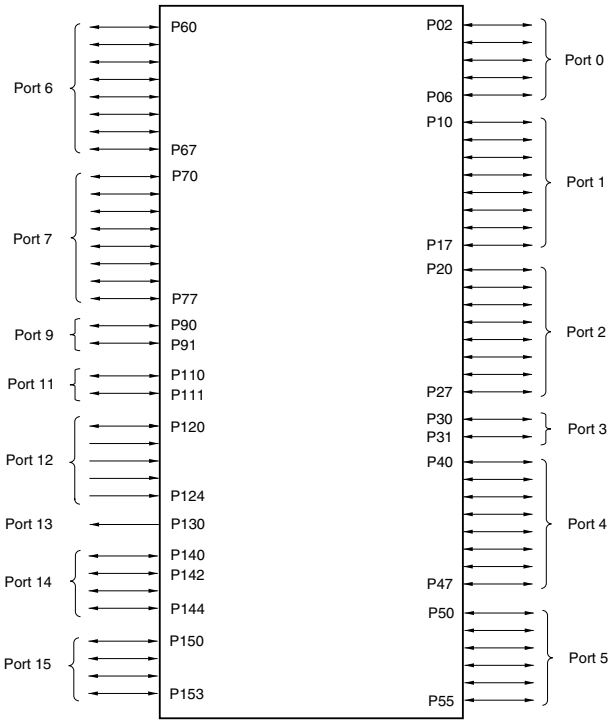


Figure 4-2. Port Types (78K0R/KG3-C)

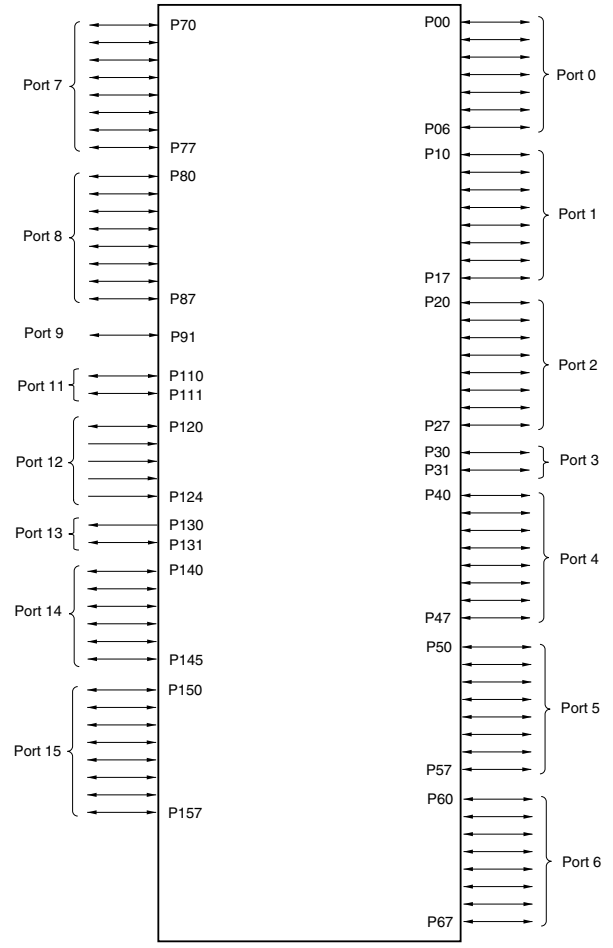


Table 4-3. Port Functions (1/3)

KF3-C	KG3-C	Function Name	I/O	Function	After Reset	Alternate Function	
						KF3-C	KG3-C
–	√	P00	I/O	Port 0. 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				$\overline{\text{SCK10}}$/SCL10	
√	√	P05				TI05/TO05	–
√	√	P06				TI06/TO06	–
√	√	P10	I/O	Port 1. 8-bit I/O port. Input of P10 and P11 can be set to TTL input buffer. Output of P10 and P12 can be set to N-ch open-drain output (V_{DD} tolerance). Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{\text{SCK00}}$	
√	√	P11				SI00/RxD0	
√	√	P12				SO00/TxD0	
√	√	P13				–	
√	√	P14				–	
√	√	P15				RTCDIV/RTCCL	
√	√	P16				TI01/TO01/INTP5	
√	√	P17				TI02/TO02	
√	√	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. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	RTC1HZ/INTP3	
√	√	P31				TI03/TO03/INTP4	
√	√	P40[Note]	I/O	Port 4. 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	TOOL0	
√	√	P41				TOOL1	
√	√	P42				TI04/TO04	
√	√	P43				$\overline{\text{SCK01}}$	
√	√	P44				SI01	
√	√	P45				SO01	
√	√	P46				RIN01	INTP1/TI05/TO05/RIN01
√	√	P47				RIN23	INTP2/RIN23

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 4-3. Port Functions (2/3)

KF3-C	KG3-C	Function Name	I/O	Function	After Reset	Alternate Function	
						KF3-C	KG3-C
√	√	P50	I/O	Port 5. 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	INTP1	–
√	√	P51				INTP2	–
√	√	P52				TO00	–
√	√	P53				TI00	–
√	√	P54				TI07/TO07	–
√	√	P55				PCLBUZ1/ INTP7	–
–	√	P56				–	
–	√	P57				–	
√	√	P60	I/O	Port 6. 8-bit I/O port. Input of P62 can be set to CEC input buffer. Output of P60 to P63 is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units. For only P62 and P64 to P67, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	SCL0	
√	√	P61				SDA0	
√	√	P62				CECIO	
√	√	P63				–	
√	√	P64				TI10/TO10	
√	√	P65				TI11/TO11	
√	√	P66				TI12/TO12	
√	√	P67				–	
√	√	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 to KR3	
√	√	P74				KR4/INTP8	
√	√	P75 to P77				KR5 to KR7	
–	√	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	–	
√	–	P90	I/O	Port 9. 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	–	
√	√	P91				ROUT	
√	√	P110	I/O	Port 11. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	CECIN	
√	√	P111				CECOUT	
√	√	P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. For only P120, Input/output can be specified. For only P120, 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	

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

KF3-C	KG3-C	Function Name	I/O	Function	After Reset	Alternate Function	
						KF3-C	KG3-C
√	√	P130	Output	Port 13. 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.	Output port	–	
–	√	P131	I/O		Input port	–	TI06/TO06
√	√	P140	I/O	Port 14. 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	
–	√	P141				–	PCLBUZ1/ INTP7
√	√	P142				$\overline{SCK20}$/SCL20	
√	√	P143				SI20/RxD2/SDA20	
√	√	P144				SO20/TxD2	
–	√	P145				–	TI07/TO07
√	√	P150 to P153	I/O	Port 15. I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI8 to ANI11	
–	√	P154 to P157				–	ANI12 to ANI15

4.2 Port Configuration

Ports include the following hardware.

Table 4-4. Port Configuration

Item	Configuration
Control registers	• 78K0R/KF3-C Port mode registers (PM0 to PM7, PM9, PM11 to PM15) Port registers (P0 to P7, P9, P11 to P15) Pull-up resistor option registers (PU0, PU1, PU3 to PU7, PU9, PU11 to PU14) • 78K0R/KG3-C Port mode registers (PM0 to PM9, PM11 to PM15) Port registers (P0 to P9, P11 to P15) Pull-up resistor option registers (PU0, PU1, PU3 to PU9, PU11 to PU14) • Common Port function register 6 (PF6) Port function register 11 (PF11) Port input mode registers (PIM0, PIM1, PIM6, PIM14) Port output mode registers (POM0, POM1, POM14) A/D port configuration register (ADPC)
Port	• 78K0R/KF3-C Total: 71 (CMOS I/O: 62, CMOS input: 4, CMOS output: 1, N-ch open drain I/O: 4) • 78K0R/KG3-C Total: 89 (CMOS I/O: 80, CMOS input: 4, CMOS output: 1, N-ch open drain I/O: 4)
Pull-up resistor	• 78K0R/KF3-C: Total: 51 • 78K0R/KG3-C: Total: 65

4.2.1 Port 0

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P00/TI00	–Note 1	√
P01/TO00	–Note 1	√
P02/SO10/TxD1	√	√
P03/SI10/RxD1/SDA10	√	√
P04/$\overline{\text{SCK10}}$/SCL10	√	√
P05/TI05/TO05	√	P05Note 2
P06/TI06/TO06	√	P06Note 2

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P06 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Input to 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, and clock I/O.

Reset signal generation sets port 0 to input mode.

Figures 4-3 to 4-8 show block diagrams of port 0.

Notes 1. TI00 and TO00 are shared with following pins, respectively, in the 78K0R/KF3-C.
P53/TI00, P52/TO00

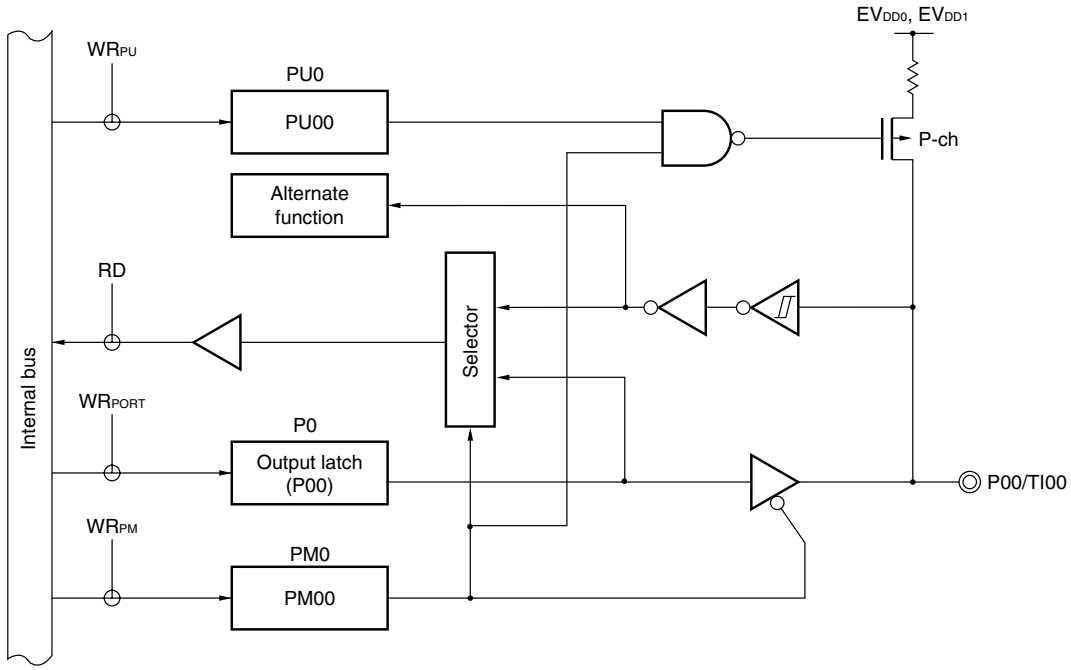
2. The 78K0R/KG3-C does not have a sharing function.
TI05/TO05 and TI06/TO06 are shared with following pins, respectively, in the 78K0R/KG3-C.
P46/INTP1/TI05/TO05/RIN01, P131/TI06/TO06

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.

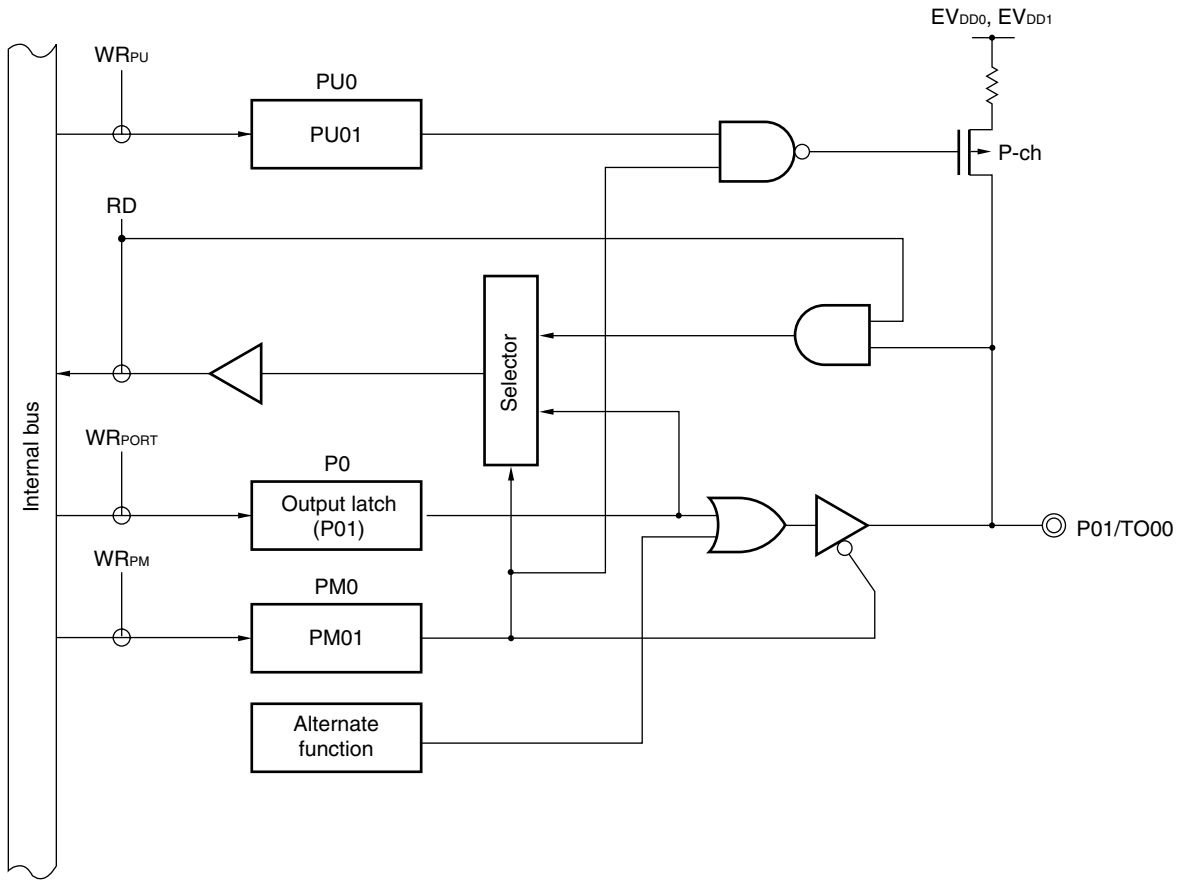
- **Table 11-7 Relationship Between Register Settings and Pins (Channel 2 of Unit 0: CSI10, UART1 Transmission, IIC10)**
- **Table 11-8 Relationship Between Register Settings and Pins (Channel 3 of Unit 0: UART1 Reception)**

Figure 4-3. Block Diagram of P00



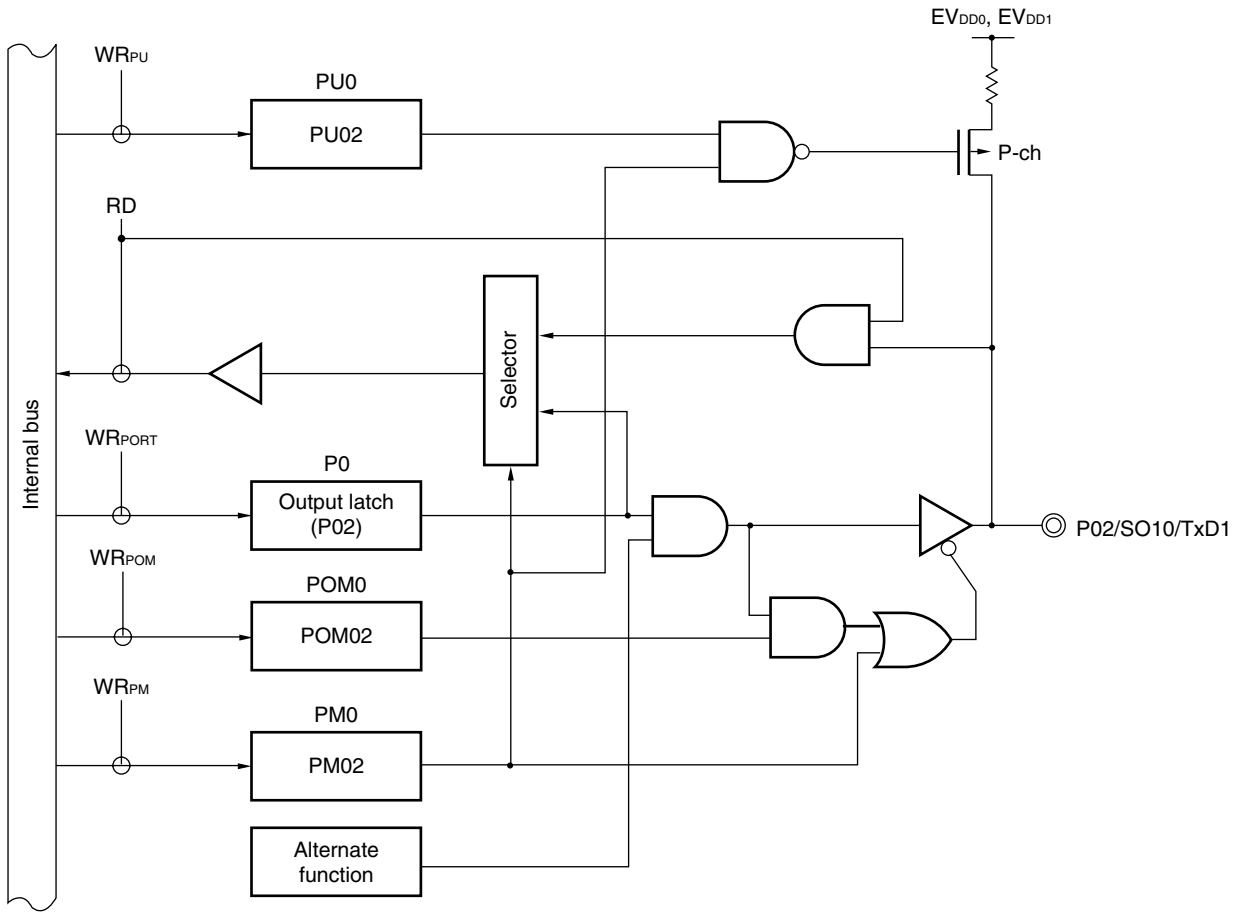
P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
RD: Read signal
WR××: Write signal

Figure 4-4. Block Diagram of P01



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
RD: Read signal
WR××: Write signal

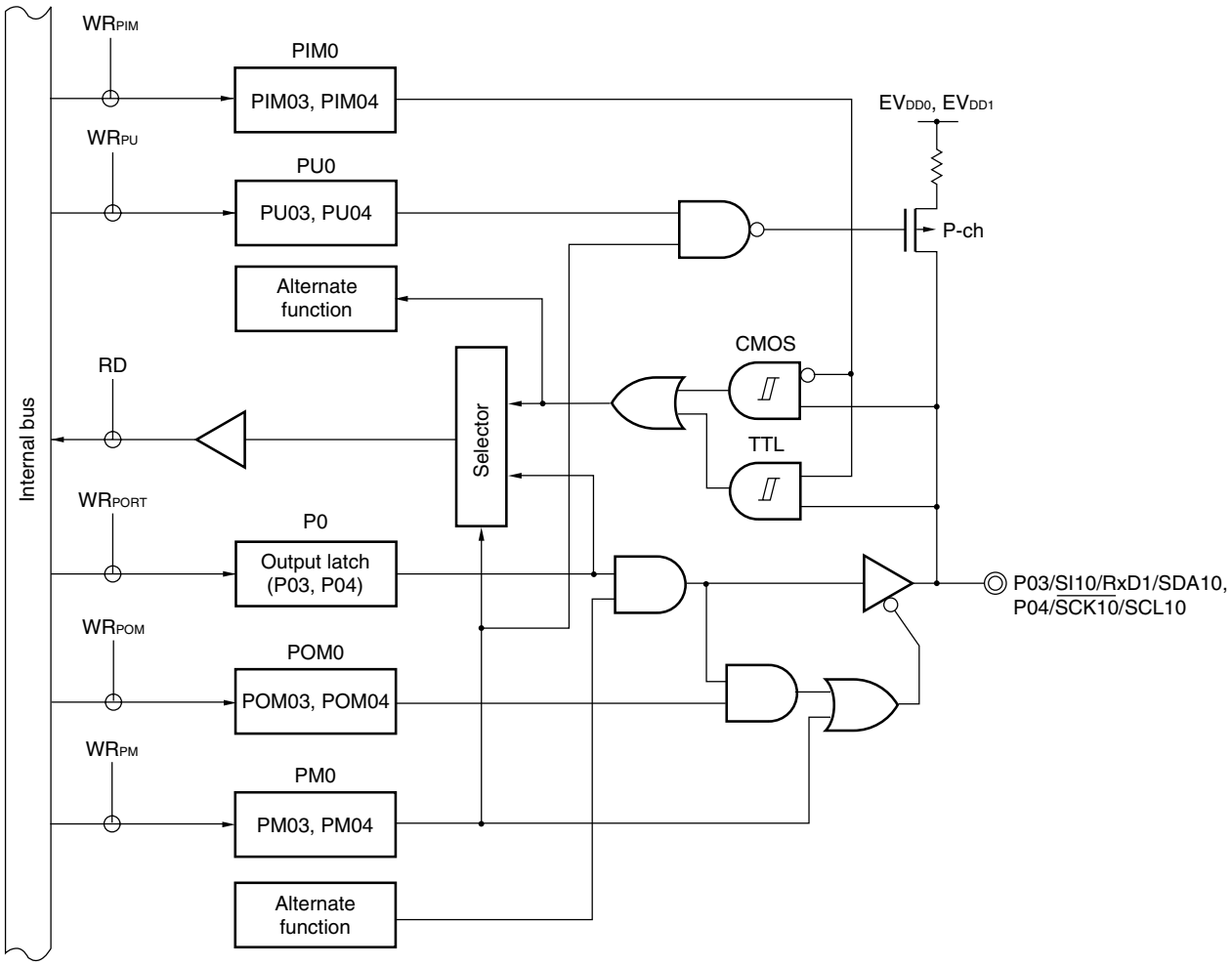
Figure 4-5. Block Diagram of P02



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××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Figure 4-6. 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××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Figure 4-7. Block Diagram of P05 and P06 (78K0R/KF3-C)

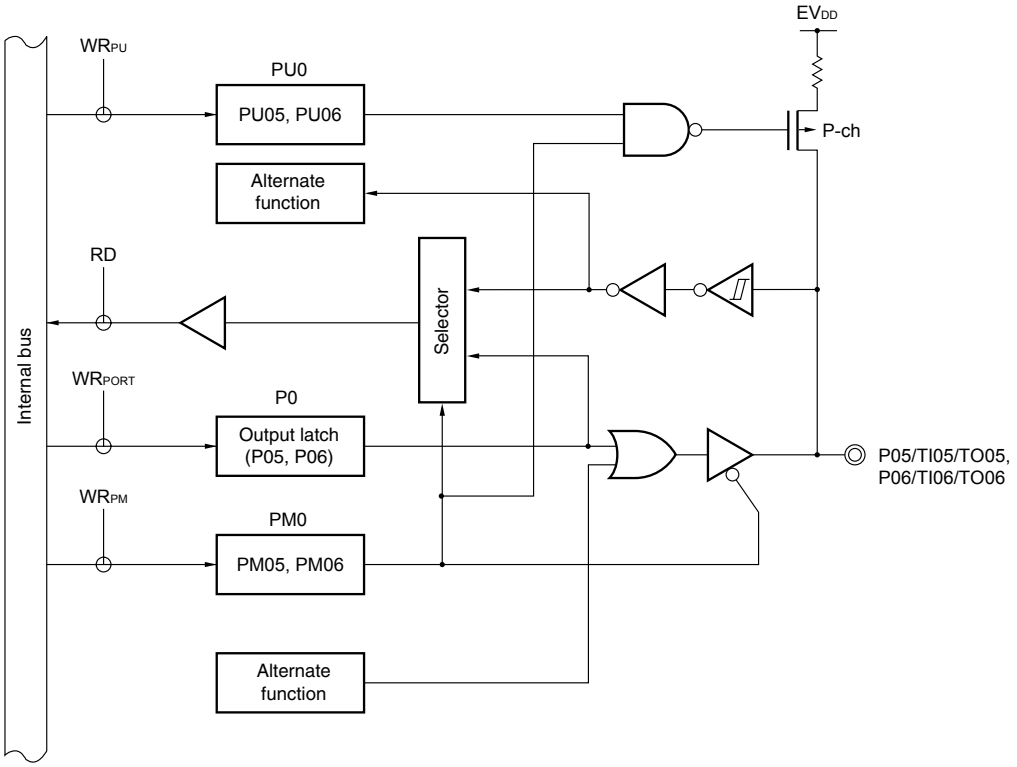
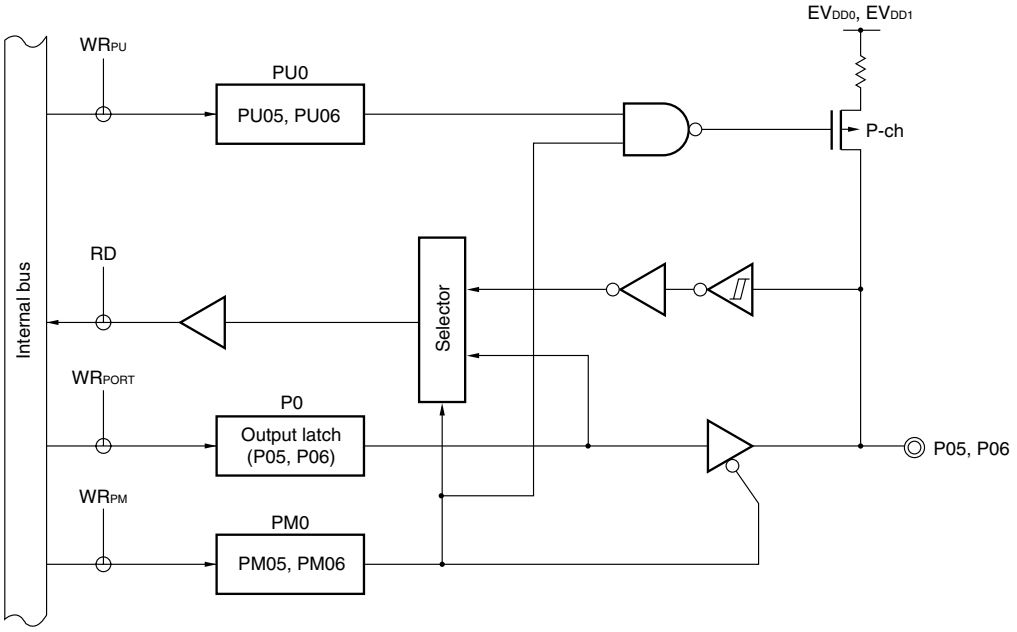


Figure 4-8. Block Diagram of P05 and P06 (78K0R/KG3-C)



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
RD: Read signal
WR××: 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).

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

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

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

Reset signal generation sets port 1 to input mode.

Figures 4-9 to 4-15 show block diagrams of port 1.

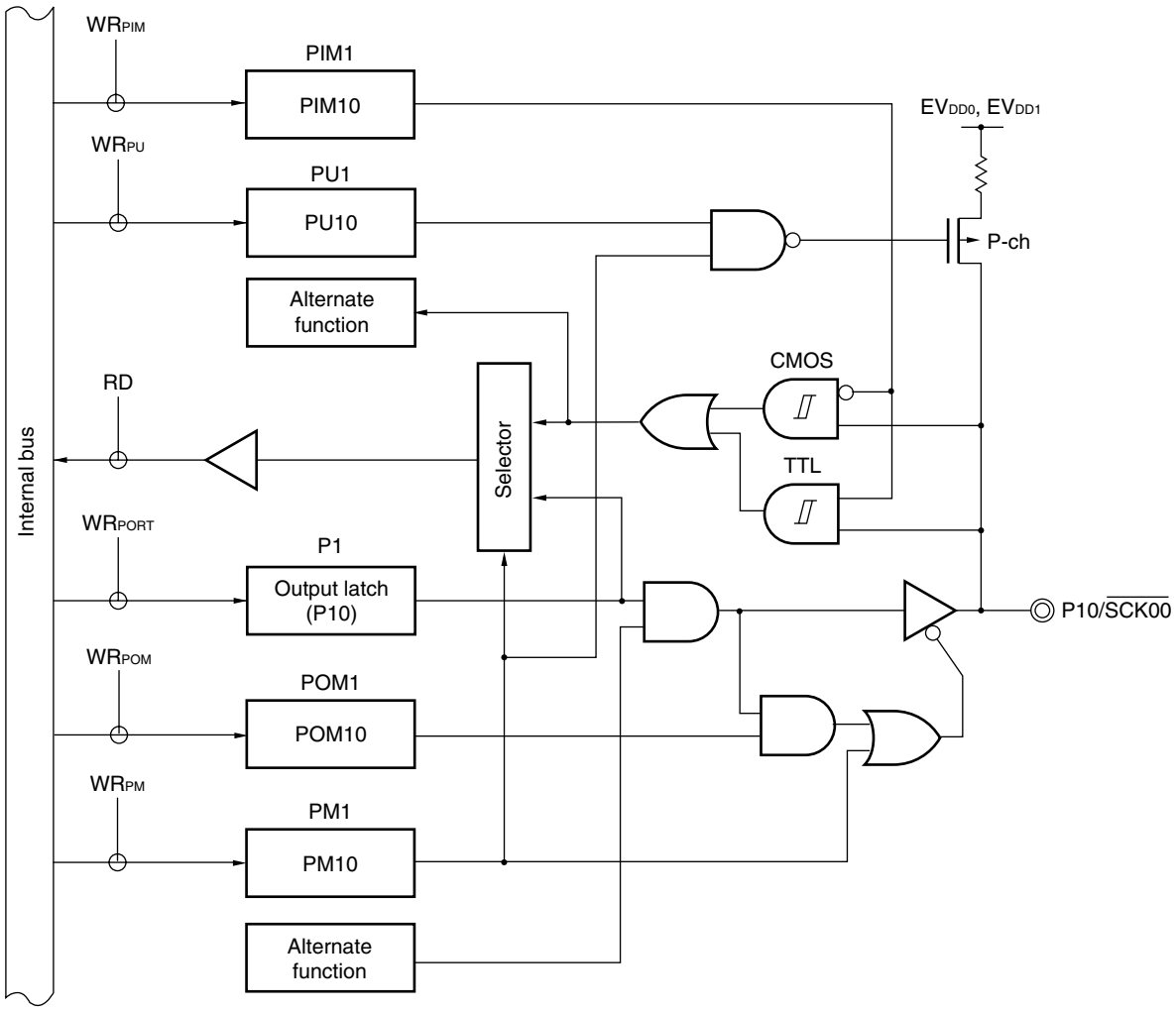
Cautions 1. To use P10/$\overline{\text{SCK00}}$, P11/SI00/RxD0, or P12/SO00/TxD0 as a general-purpose port, note the serial array unit setting. For details, refer to the following tables.

- **Table 11-5 Relationship Between Register Settings and Pins (Channel 0 of Unit 0: CSI00, UART0 Transmission)**
- **Table 11-6 Relationship Between Register Settings and Pins (Channel 1 of Unit 0: CSI01, UART0 Reception)**

2. To use P16/TI01/TO01/INTP5 or P17/TI02/TO02 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 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.

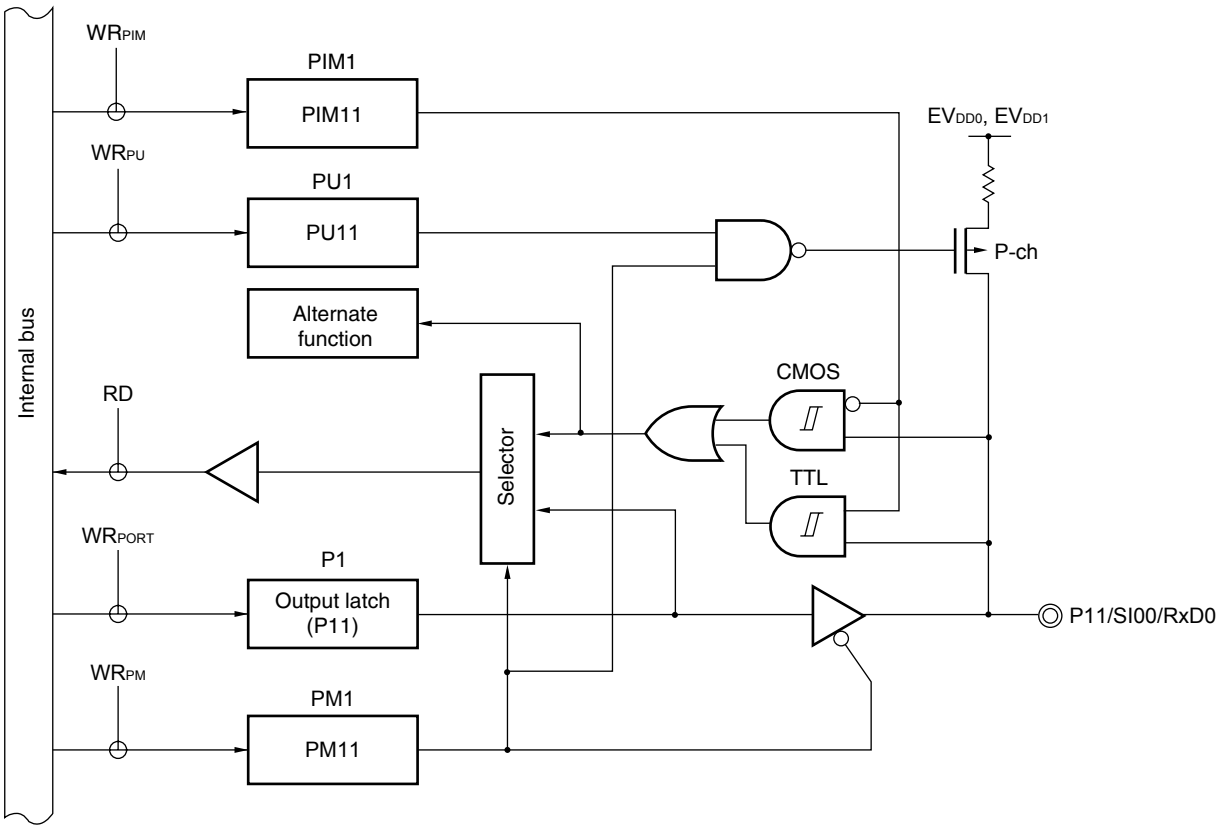
Figure 4-9. Block Diagram of P10



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
RD: Read signal
WR××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

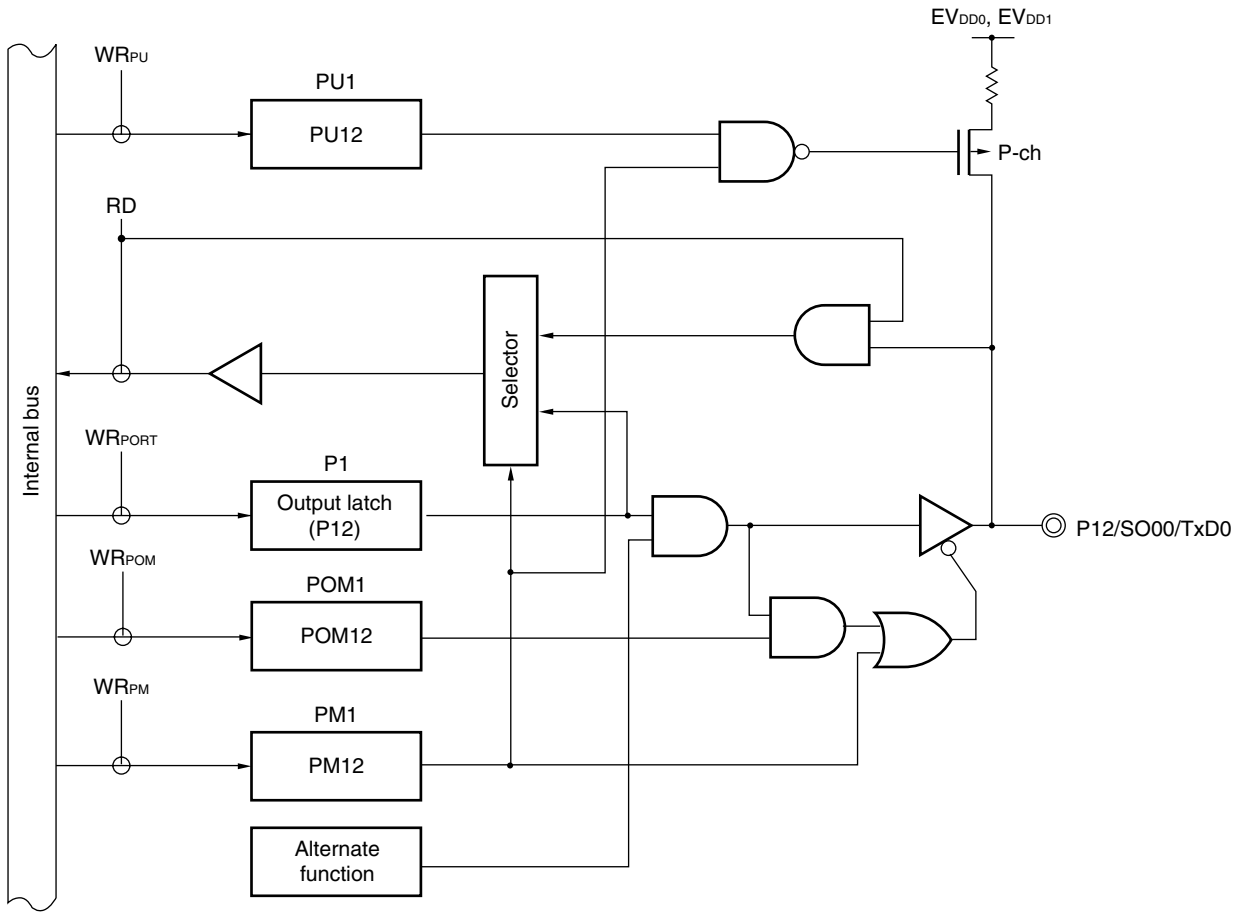
Figure 4-10. Block Diagram of P11



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
PIM1: Port input mode register 1
RD: Read signal
WR××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

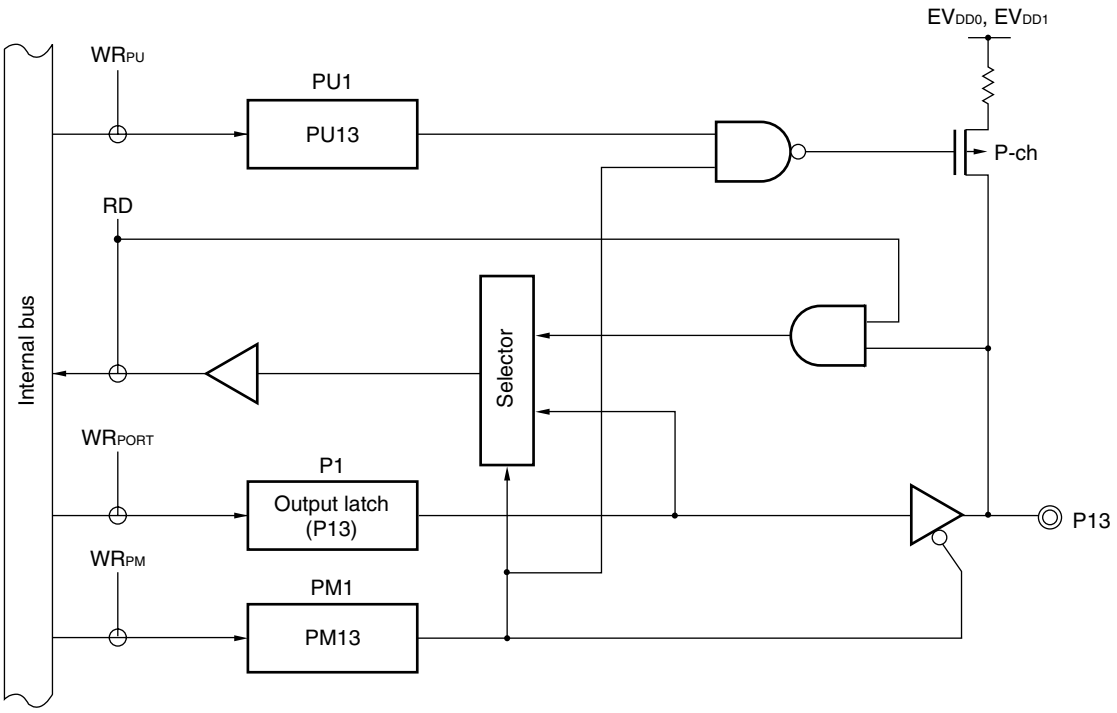
Figure 4-11. Block Diagram of P12



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
POM1: Port output mode register 1
RD: Read signal
WR××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

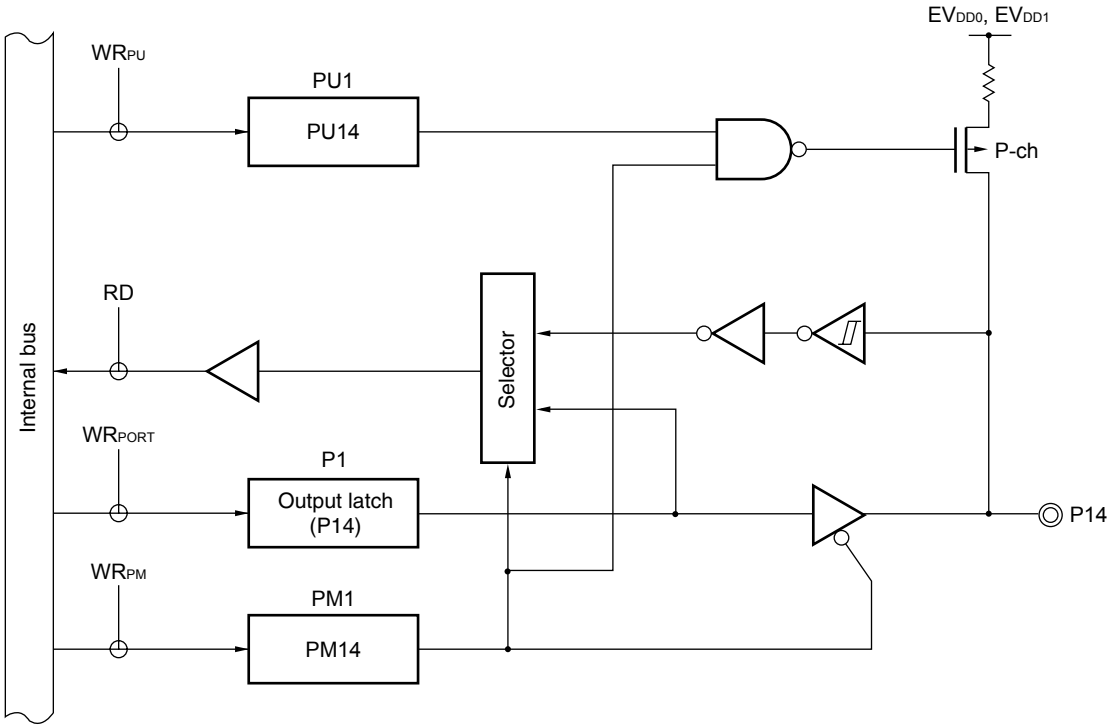
Figure 4-12. Block Diagram of P13



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

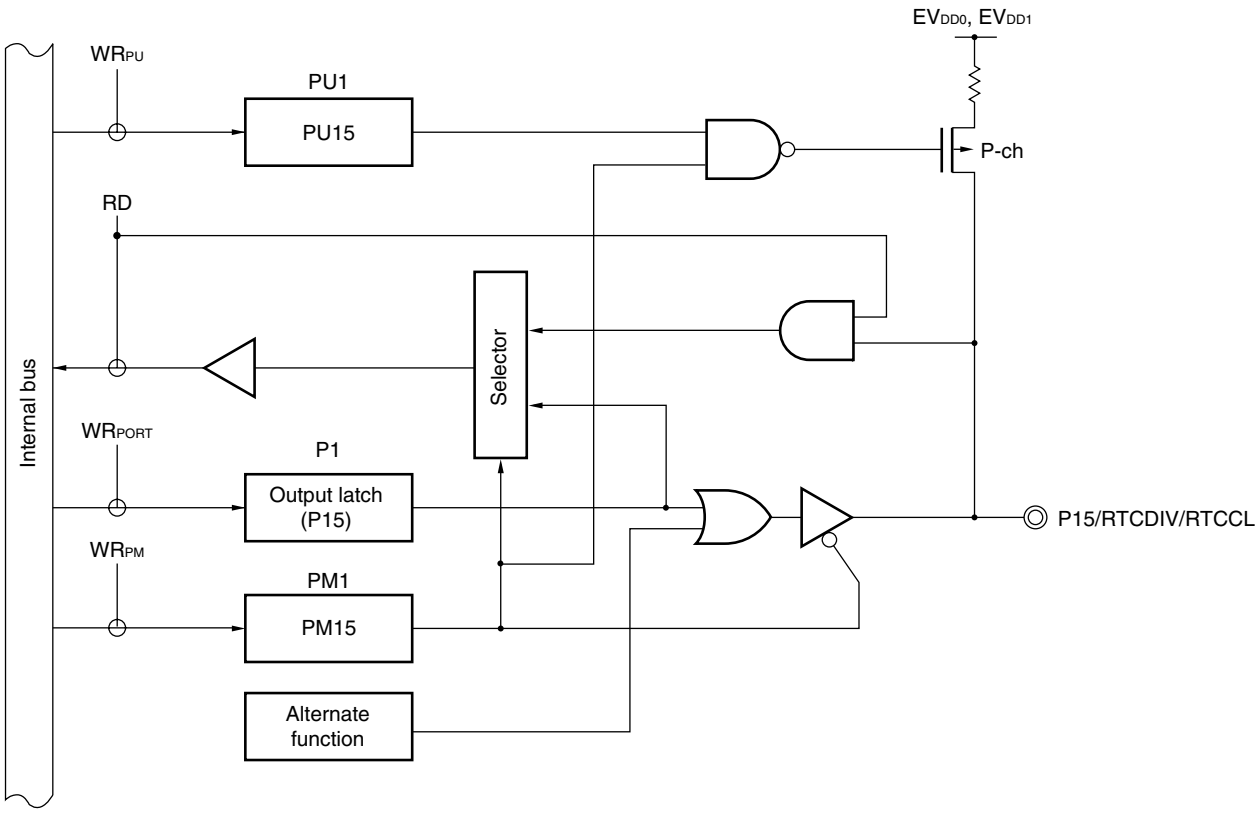
Figure 4-13. Block Diagram of P14



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

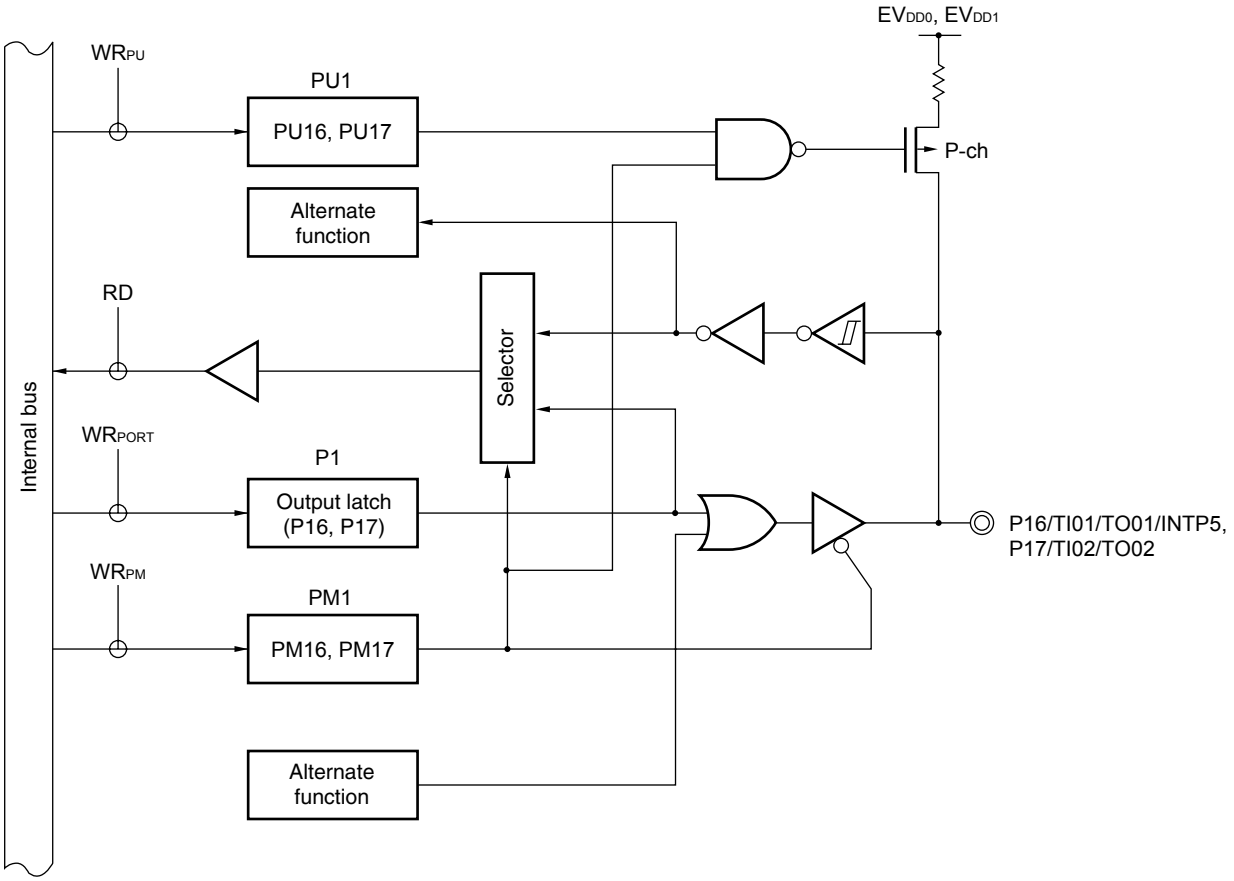
Figure 4-14. Block Diagram of P15



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Figure 4-15. 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××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

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 4-5. Setting Functions of P20/ANI0 to P27/ANI7 Pins

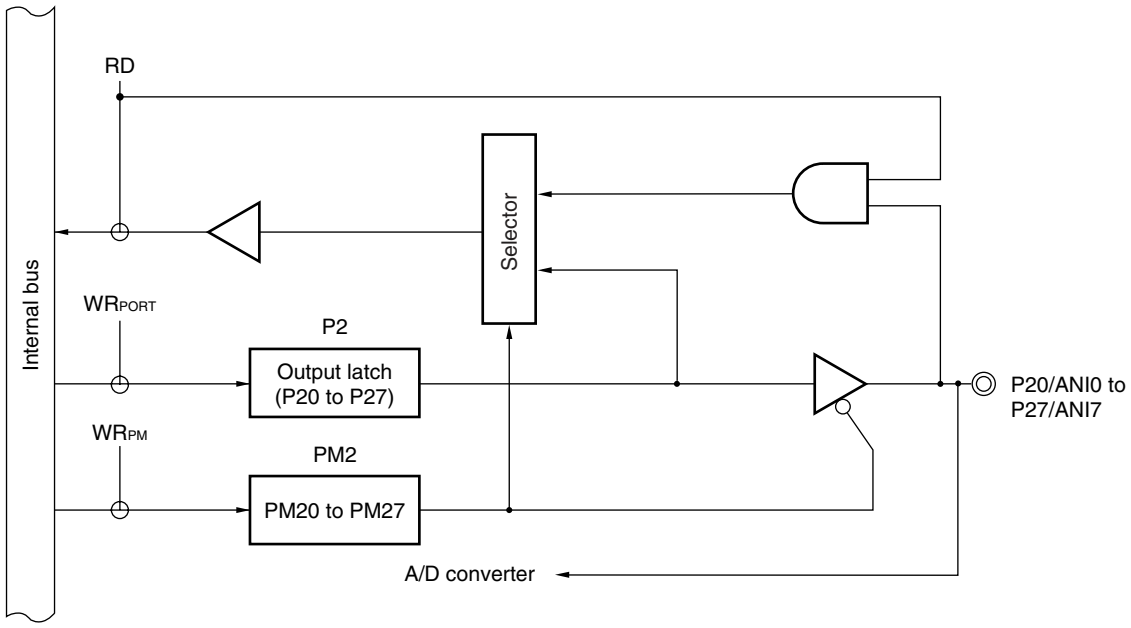
ADPC	PM2	ADS	P20/ANI0 to P27/ANI7 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
	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

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

Figure 4-16 shows a block diagram of port 2.

Caution See 2.2.16 AV_{REF} for the voltage to be applied to the AV_{REF} pin when using port 2 as a digital I/O.

Figure 4-16. Block Diagram of P20 to P27



P2: Port register 2
PM2: Port mode register 2
RD: Read signal
WR××: Write signal

4.2.4 Port 3

Port 3 is a 2-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 and P31 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

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

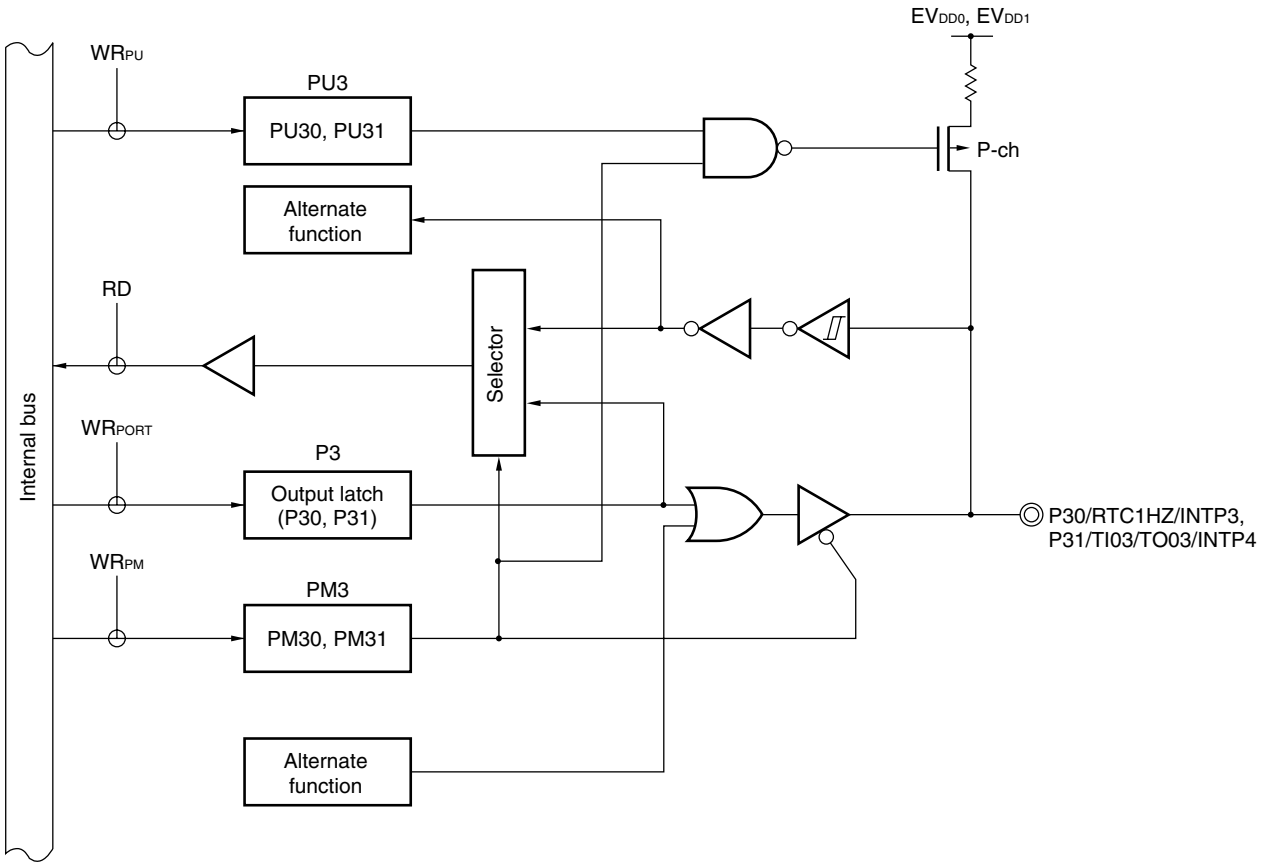
Reset signal generation sets port 3 to input mode.

Figure 4-17 shows block a diagram 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 its default status setting.

Figure 4-17. 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××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

4.2.5 Port 4

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P40/TOOL0	√	√
P41/TOOL1	√	√
P42/TI04/TO04	√	√
P43/$\overline{\text{SCK01}}$	√	√
P44/SI01	√	√
P45/SO01	√	√
P46/INTP1/TI05/TO05/RIN01	P46/RIN01 Note 1	√
P47/INTP2/RIN23	P47/RIN23 Note 1	√

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)Note 2.

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, timer I/O, and remote control receive data input.

Reset signal generation sets port 4 to input mode.

Figures 4-18 to 4-23 show block diagrams of port 4.

Notes 1. INTP1, TI05/TO05, and INTP2 are shared with following pins, respectively, in the 78K0R/KF3-C.
P50/INTP1, P05/TI05/TO05, P51/INTP2

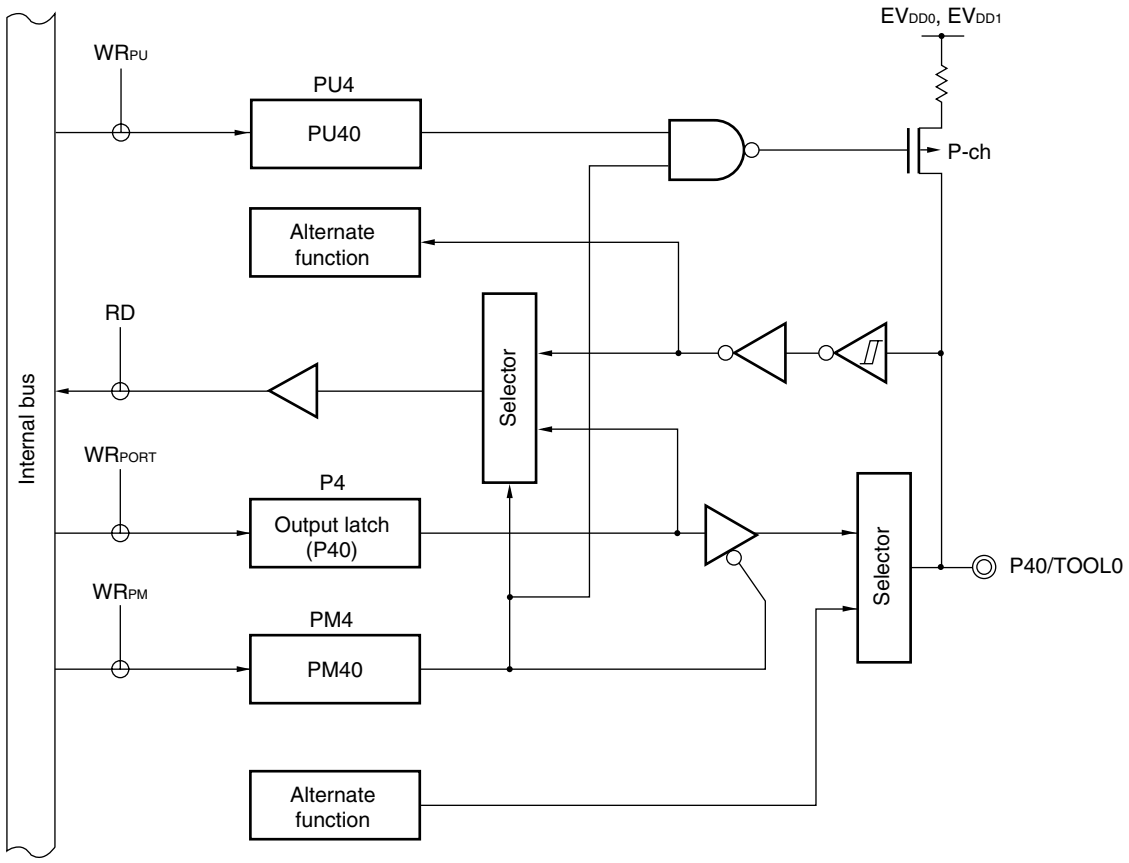
2. When a tool is connected, the P40 and P41 pins cannot be connected to a pull-up resistor.

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/$\overline{\text{SCK01}}$, P44/SI01, or P45/SO01 as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 11-6 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/RIN01 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.

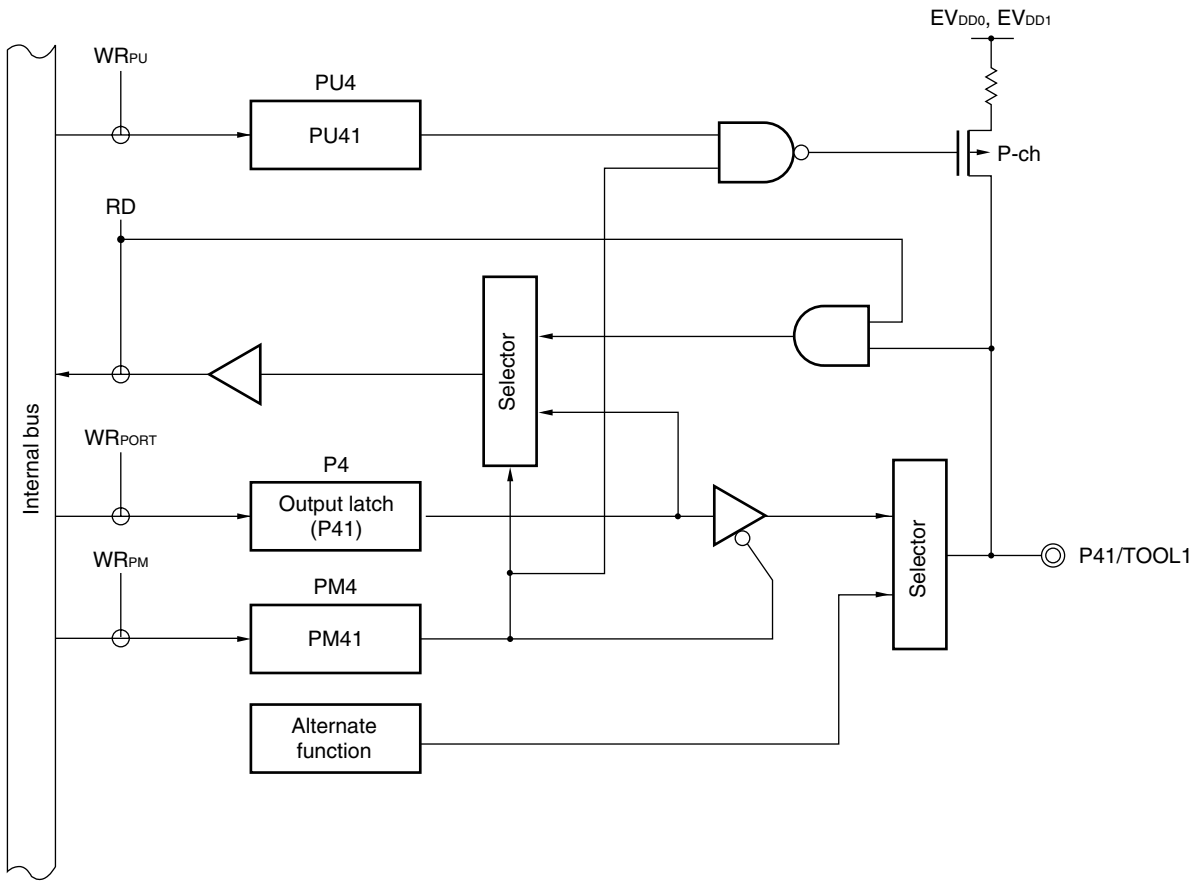
Figure 4-18. Block Diagram of P40



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

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

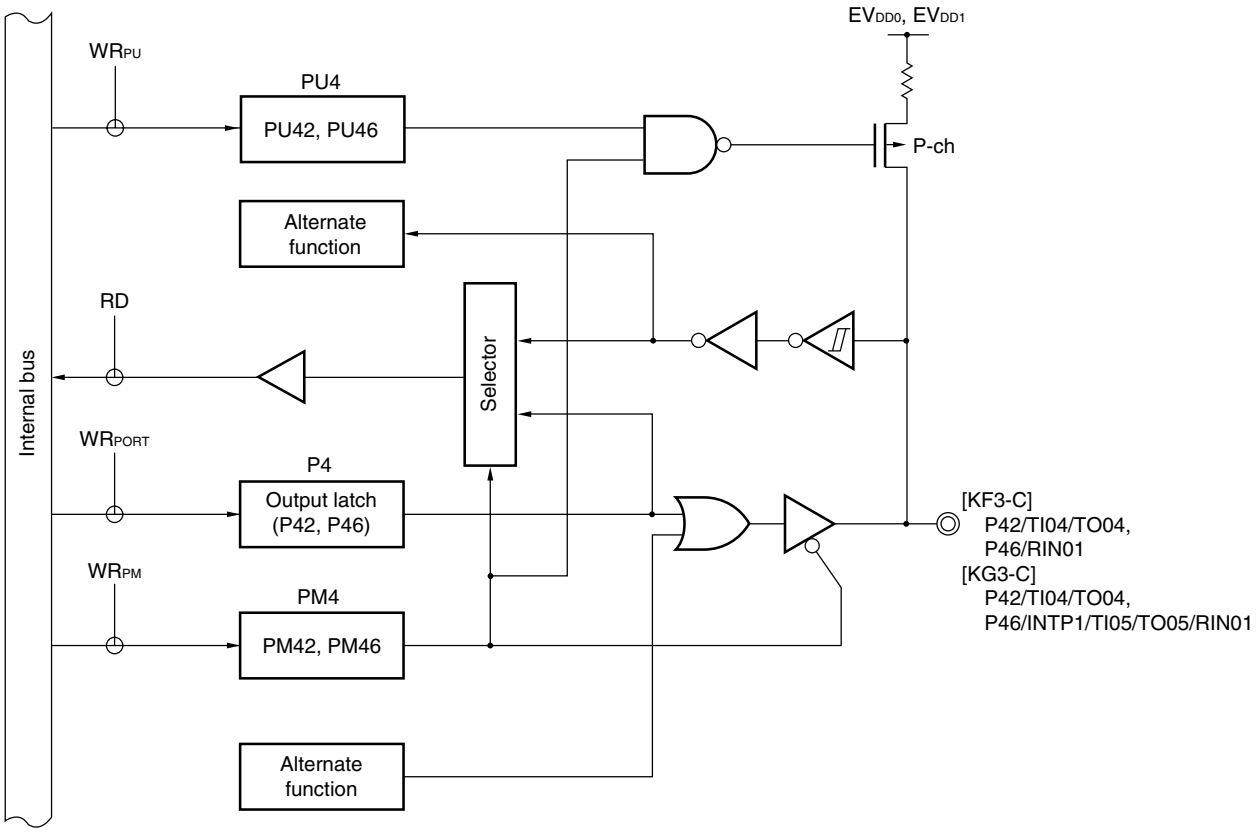
Figure 4-19. Block Diagram of P41



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

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

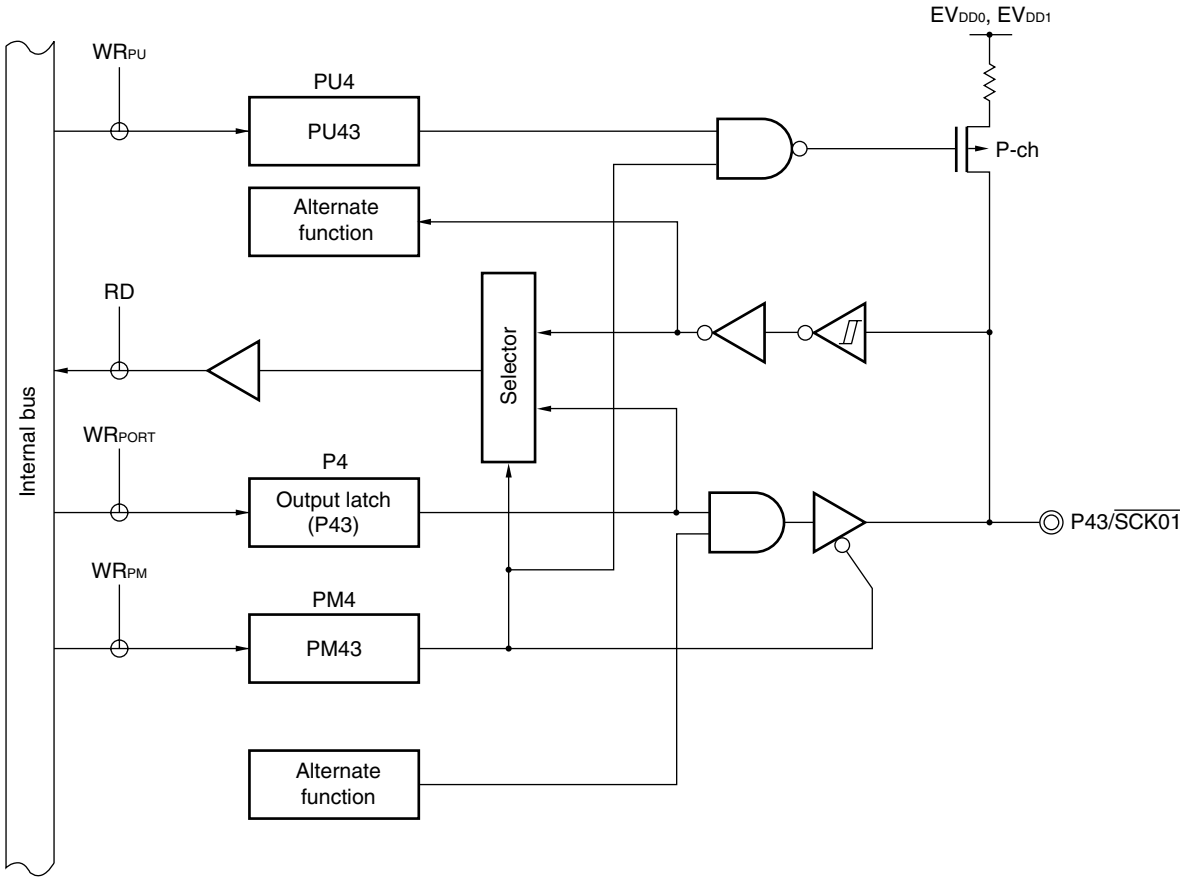
Figure 4-20. Block Diagram of P42, P46



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

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

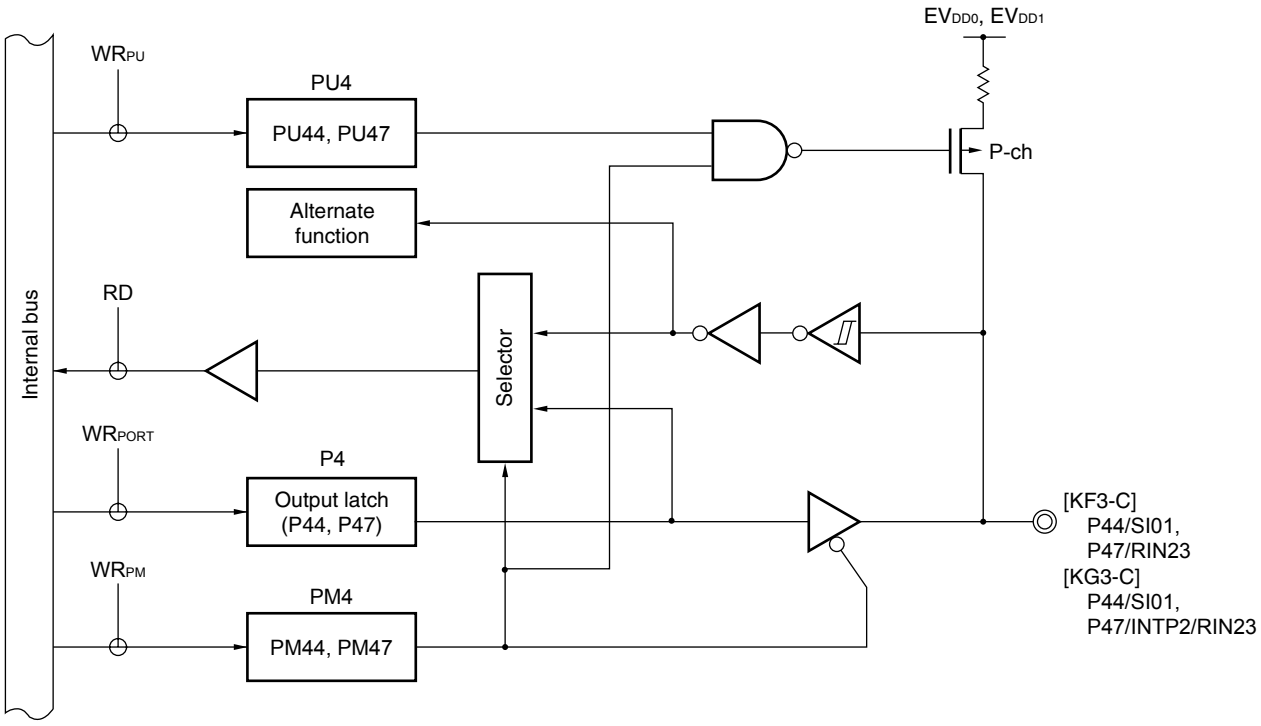
Figure 4-21. Block Diagram of P43



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

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

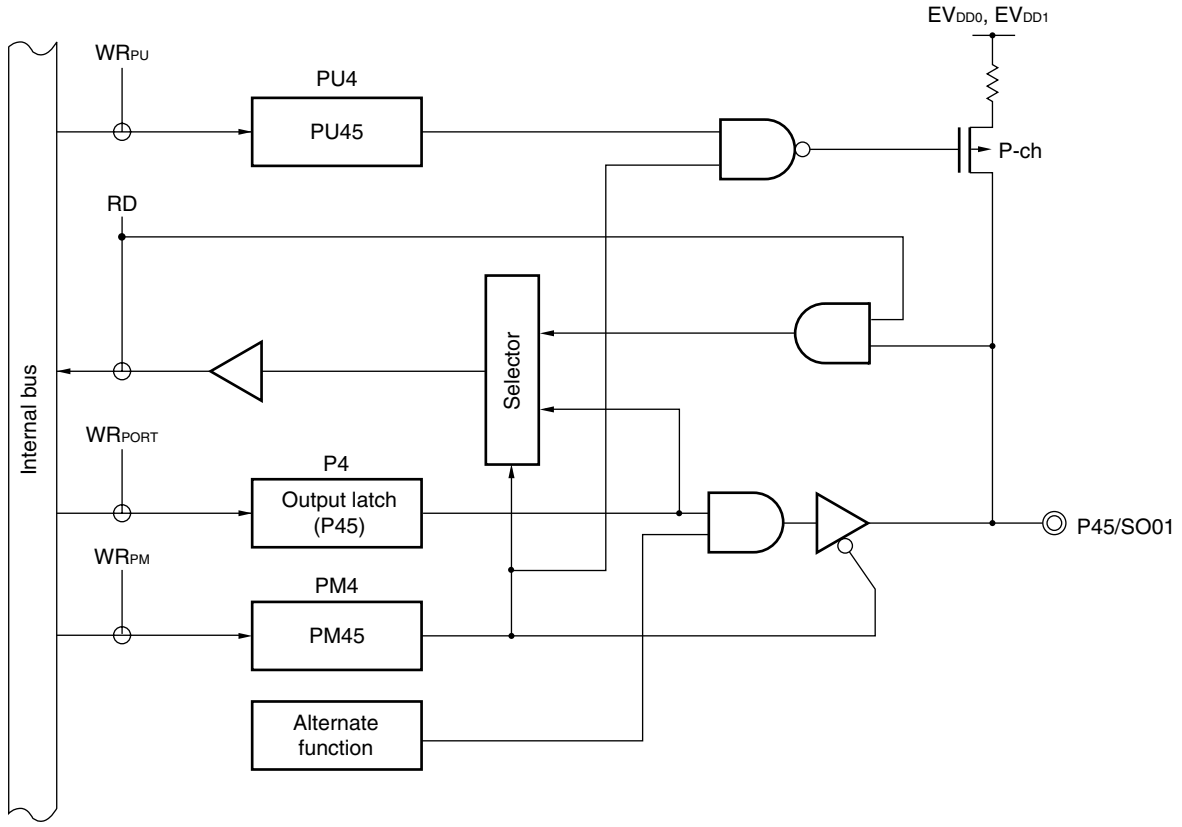
Figure 4-22. Block Diagram of P44, P47



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

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Figure 4-23. Block Diagram of P45



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

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

4.2.6 Port 5

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P50/INTP1	√	P50[Note]
P51/INTP2	√	P51[Note]
P52/TO00	√	P52[Note]
P53/TI00	√	P53[Note]
P54/TI07/TO07	√	P54[Note]
P55/PCLBUZ1/INTP7	√	P55[Note]
P56	–	√
P57	–	√

Note The 78K0R/KG3-C does not have a sharing function.
INTP1, INTP2, TO00, TI00, TI07/TO07, and PCLBUZ1/INTP7 are shared with following pins, respectively, in the 78K0R/KG3-C.
P46/INTP1/TI05/TO05/RIN01, P47/INTP2/RIN23, P01/TO00, P00/TI00, P145/TI07/TO07, P141/PCLBUZ1/INTP7

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

Reset signal generation sets port 5 to input mode.

Figures 4-24 and 4-28 show block diagrams of port 5.

Cautions 1. To use P52/TO00 or P54/TI07/TO07 as a general-purpose port, set bits 0 and 7 (TO00, TO07) of timer output register 0 (TO0) and bits 0 and 7 (TOE00, TOE07) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

2. To use P55/PCLBUZ1/INTP7 as a general-purpose port, set bit 7 of clock output select register 1 (CKS1) to "0", which is the same as their default status settings.

(1) 78K0R/KF3-C (1/2)

Figure 4-24. Block Diagram of P50, P51, and P53

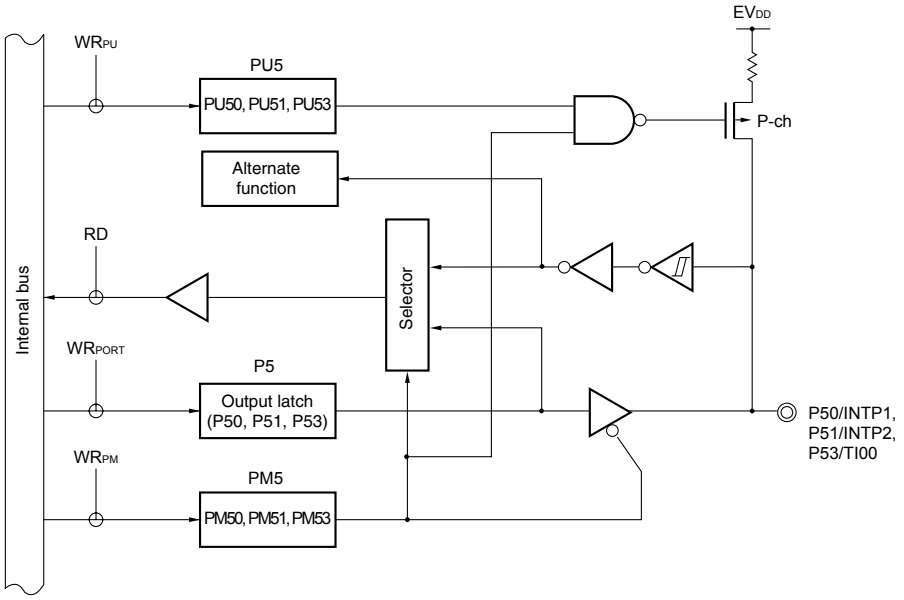
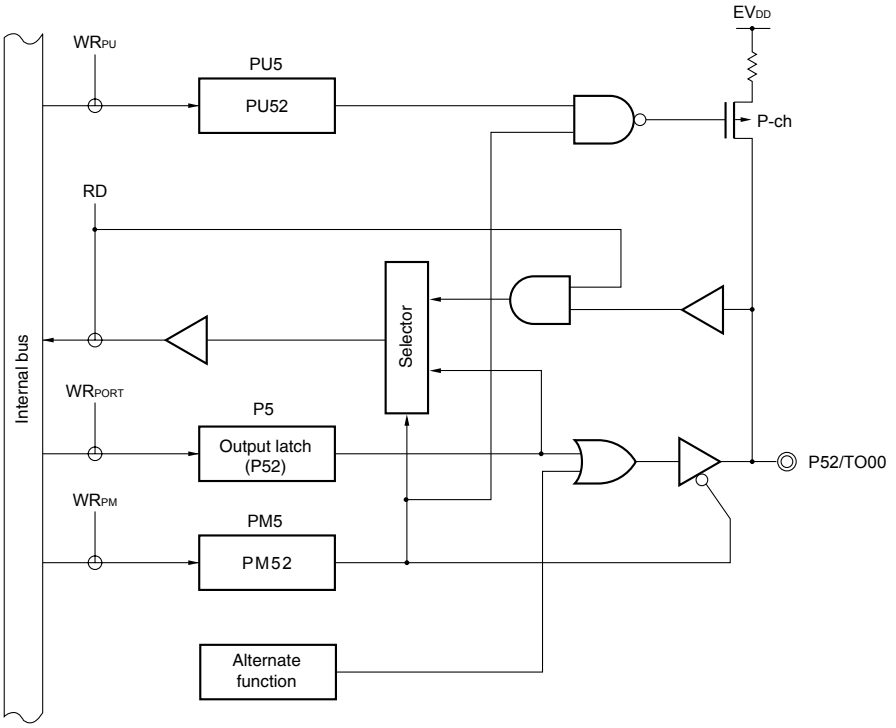


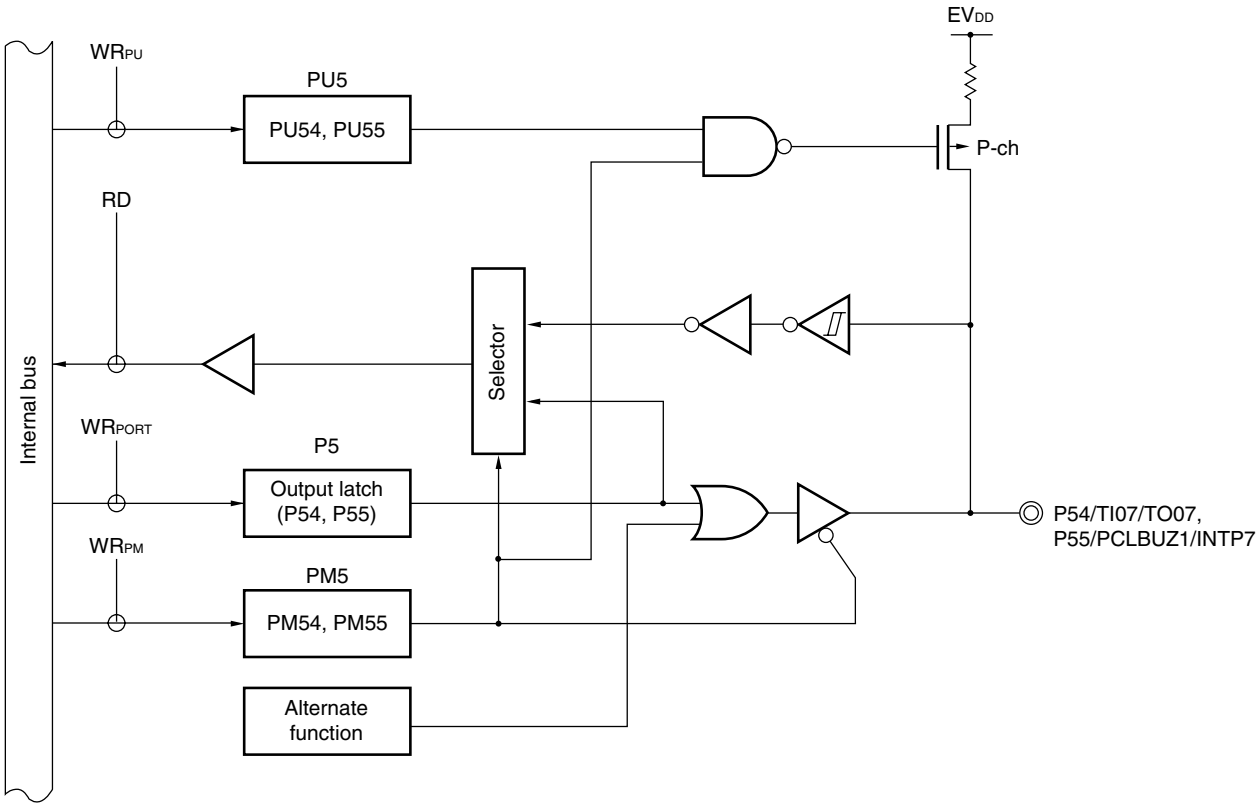
Figure 4-25. Block Diagram of P52



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
RD: Read signal
WR××: Write signal

(1) 78K0R/KF3-C (2/2)

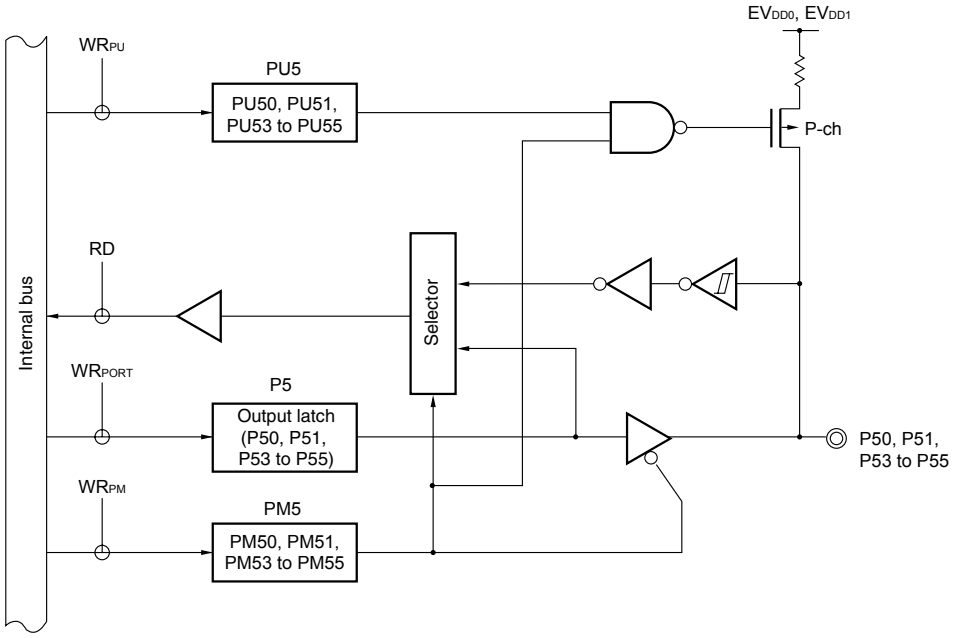
Figure 4-26. Block Diagram of P54 and P55



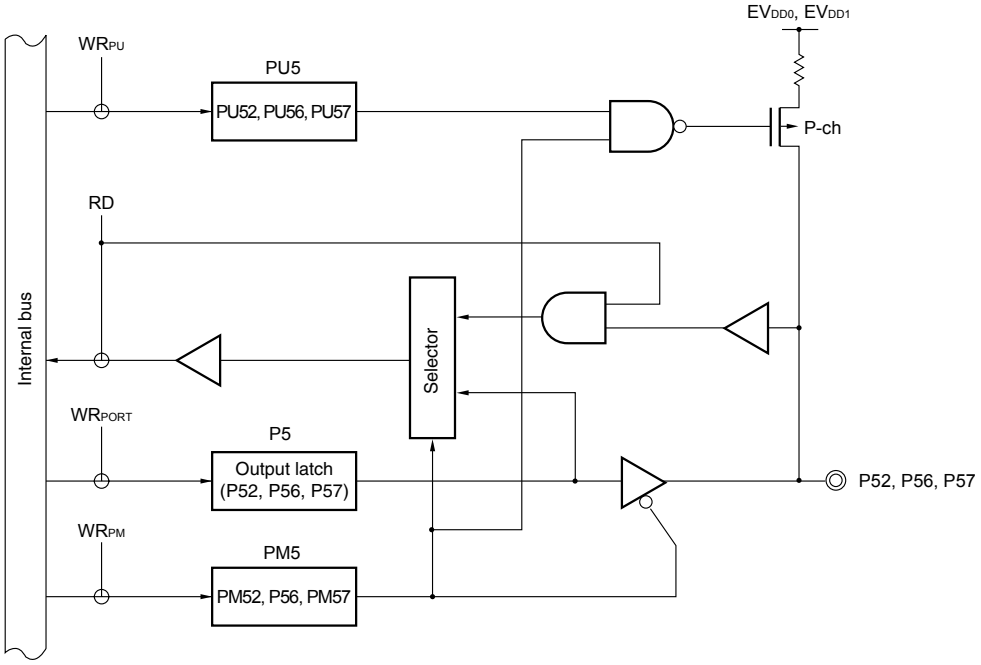
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
RD: Read signal
WR××: Write signal

(2) 78K0R/KG3-C

Figure 4-27. Block Diagram of P50, P51, and P53 to P55



igure 4-28. Block Diagram of P52, P56, and P57



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
RD: Read signal
WR××: Write signal

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 P62 and 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, serial data I/O for CEC, and timer I/O.

Input to the P62 pin can be specified through a normal input buffer or a CEC input buffer, using port function register 6 (PF6) and port input mode register 6 (PIM6) (see **Figure 4-59**).

Only the P62 pin can be connected to an internal diode and pull-up resistor by setting port function register 6 (PF6) and pull-up resistor option register 6 (PU6) (see **Figure 4-59**).

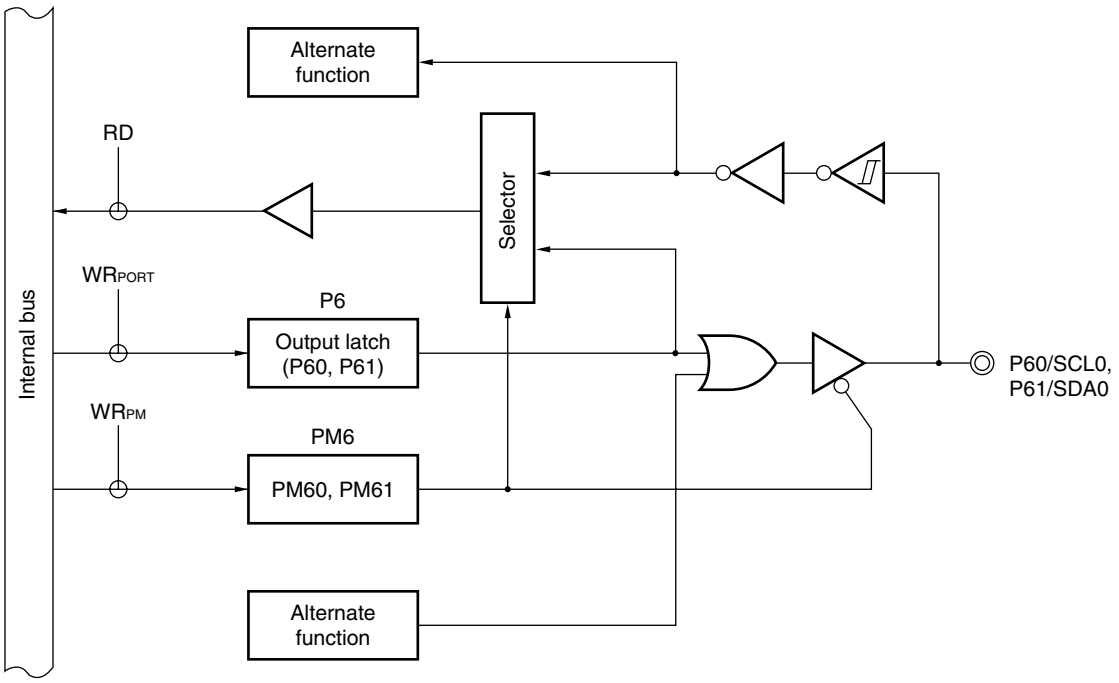
Reset signal generation sets port 6 to input mode.

Figures 4-29 to 4-33 show block diagrams of port 6.

Cautions 1. Stop the operation of serial interface IICA when using P60/SCL0 and P61/SDA0 as general-purpose ports.

2. To use P64/TI10/TO10 to P66/TI12/TO12 as a general-purpose port, set bits 0 to 2 (TO10 to TO12) of timer output register 1 (TO1) and bits 0 to 2 (TOE10 to TOE12) of timer output enable register 1 (TOE1) to "0", which is the same as their default status setting.

Figure 4-29. Block Diagram of P60 and P61



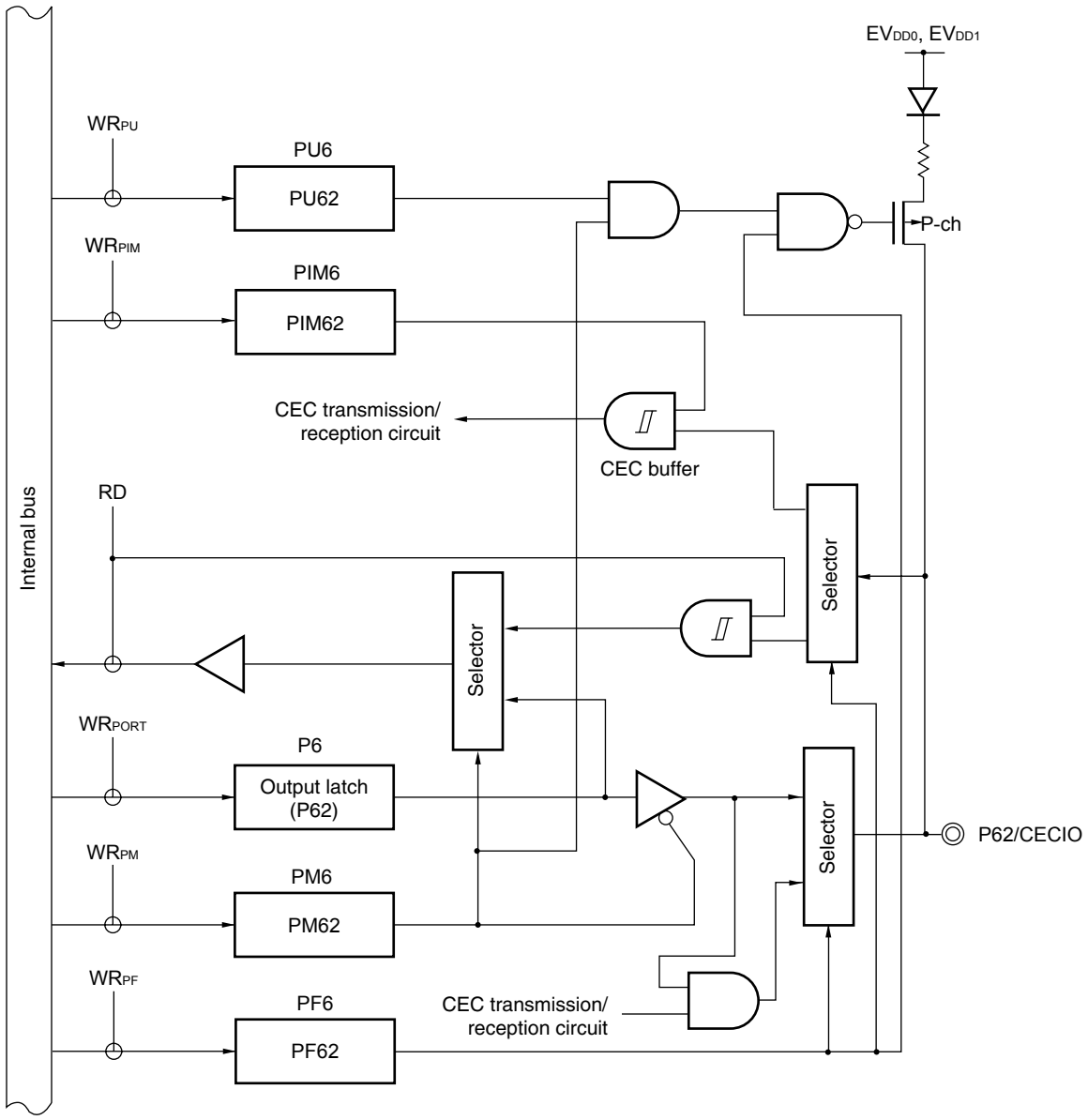
P6: Port register 6

PM6: Port mode register 6

RD: Read signal

WR××: Write signal

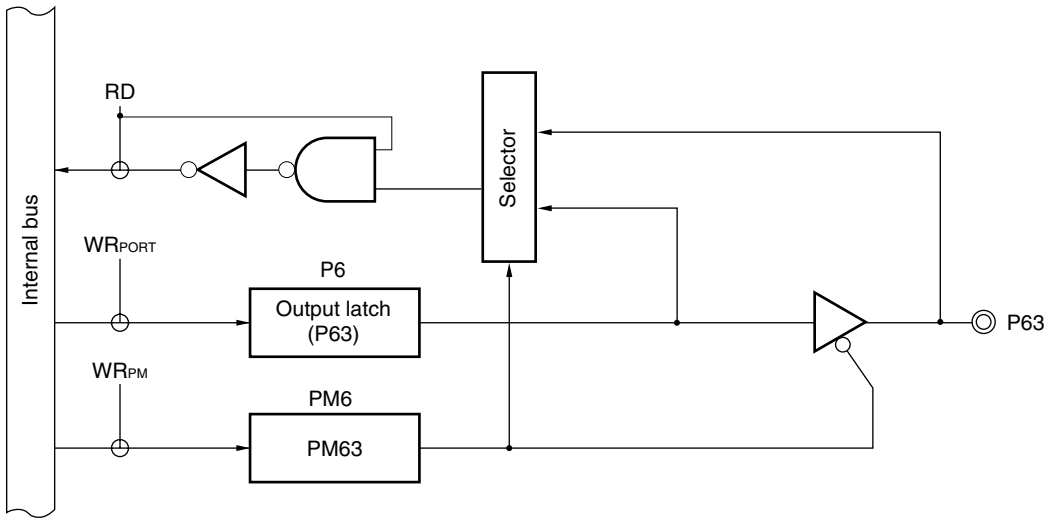
Figure 4-30. Block Diagram of P62



P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
PIM6: Port input mode register 6
PF6: Port function register 6
RD: Read signal
WR××: Write signal

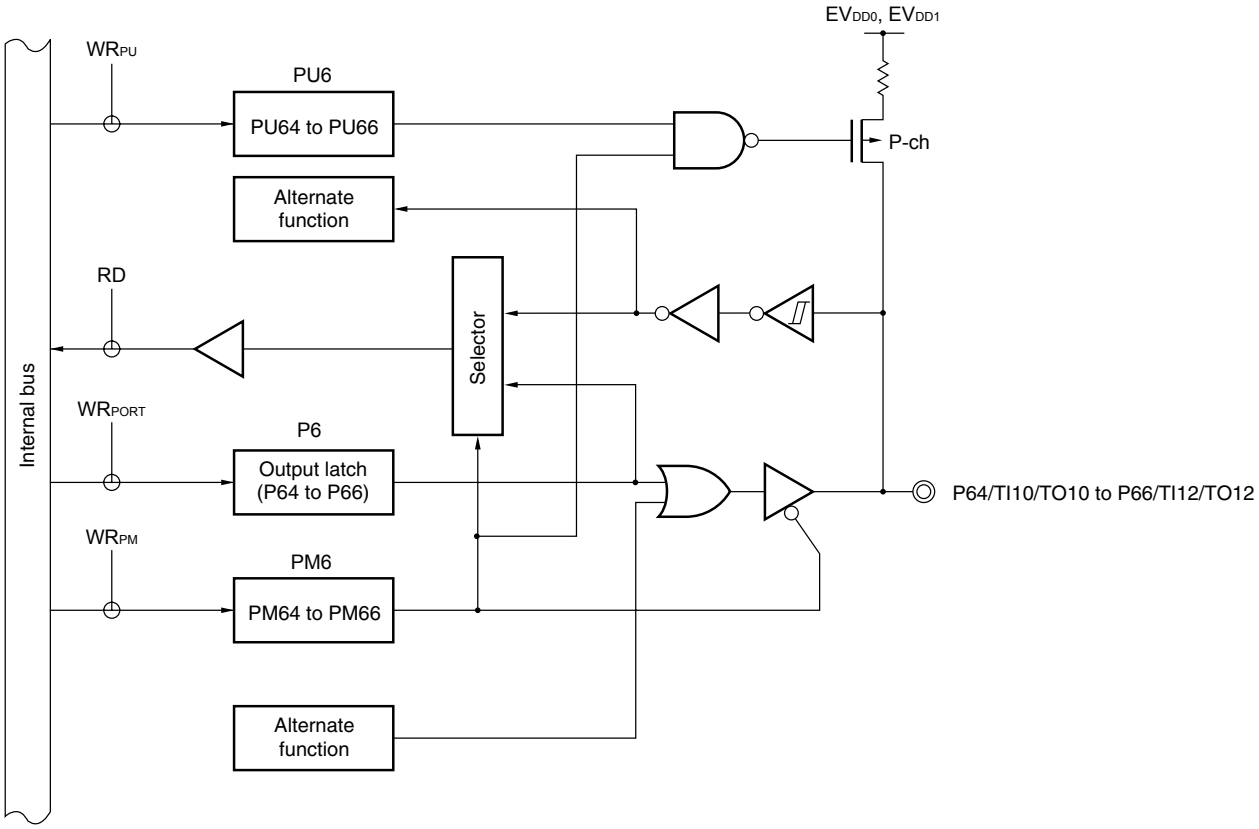
Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Figure 4-31. Block Diagram of P63



P6: Port register 6
PM6: Port mode register 6
RD: Read signal
WR××: Write signal

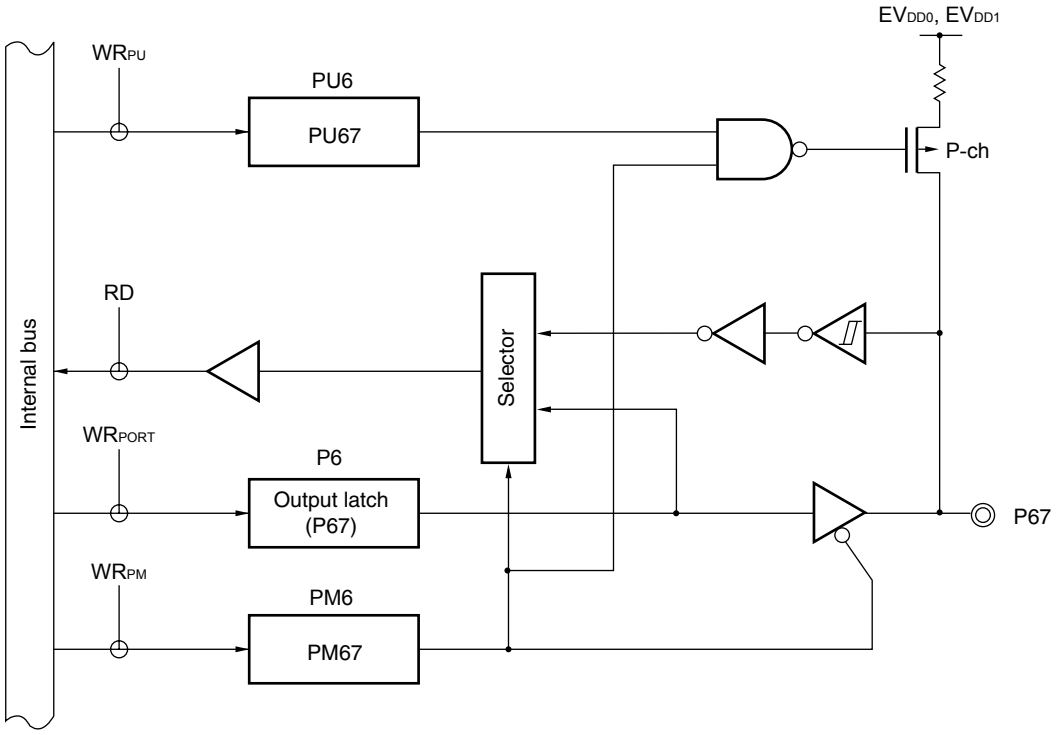
Figure 4-32. Block Diagram of P64 to P66



P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
RD: Read signal
WR××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Figure 4-33. Block Diagram of P67



P6: Port register 6
PU6: Pull-up resistor option register 6
PM6: Port mode register 6
RD: Read signal
WR××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

4.2.8 Port 7

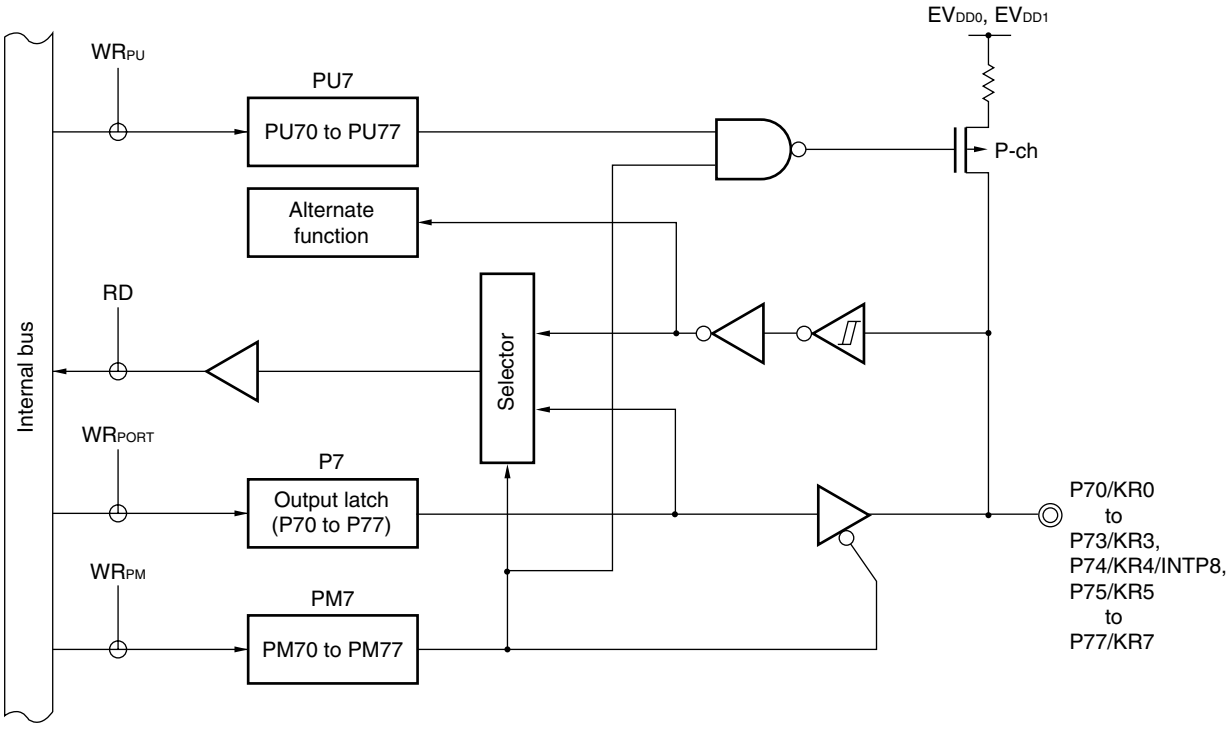
Port 7 is an 8-bit I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

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

Reset signal generation sets port 7 to input mode.

Figure 4-34 shows a block diagram of port 7.

Figure 4-34. 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××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

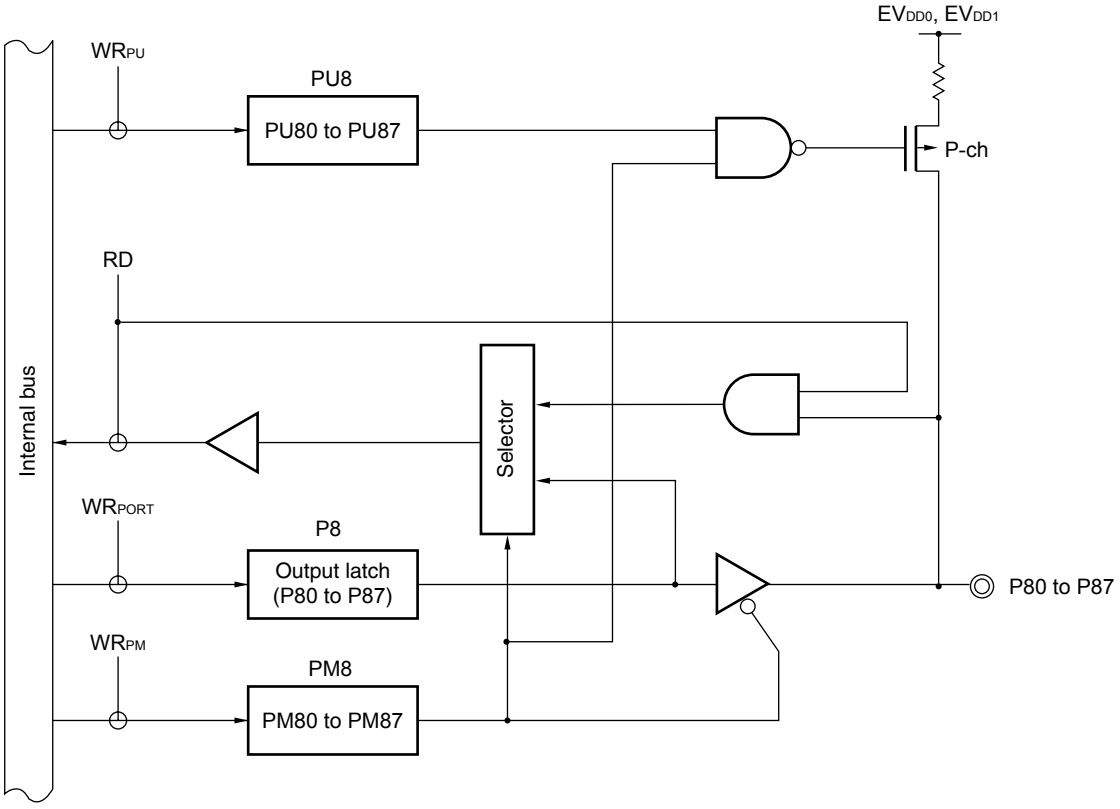
4.2.9 Port 8 (78K0R/KG3-C only)

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).

Reset signal generation sets port 8 to input mode.

Figure 4-35 shows a block diagram of port 8.

Figure 4-35. 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××: Write signal

4.2.10 Port 9

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P90	√	–
P91/ROUT	√	√

P90 and P91 are an I/O port with an output latch. Port 9 can be set to the input mode or output mode using port mode register 9 (PM9). When the P90 and P91 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).

P91 can also be used for remote control receive data output.

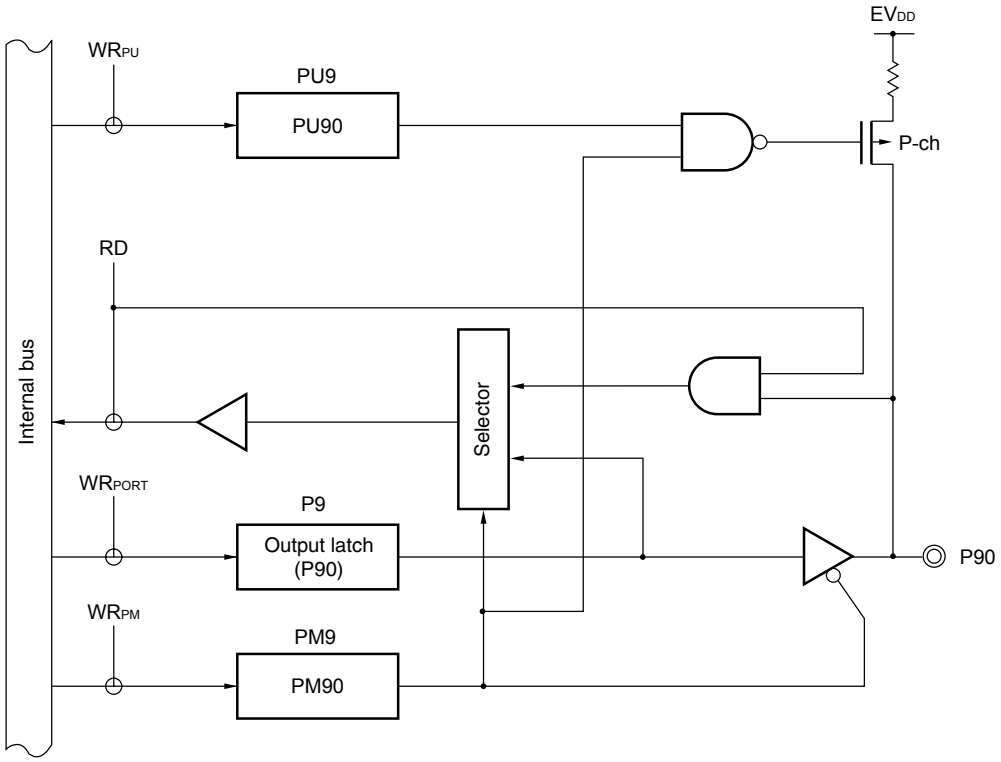
By specifying a setting using the remote controller receive data through control register (RMSW), it is possible to output the remote controller receive data input from the RIN01 and RIN23 pins from these pins without noise elimination or decoding (For details, see 14.3 (5) Remote controller receive data through control register (RMSW)).

Reset signal generation sets P90 and P91 to input mode.

Figures 4-36 and 4-37 show block diagrams of port 9.

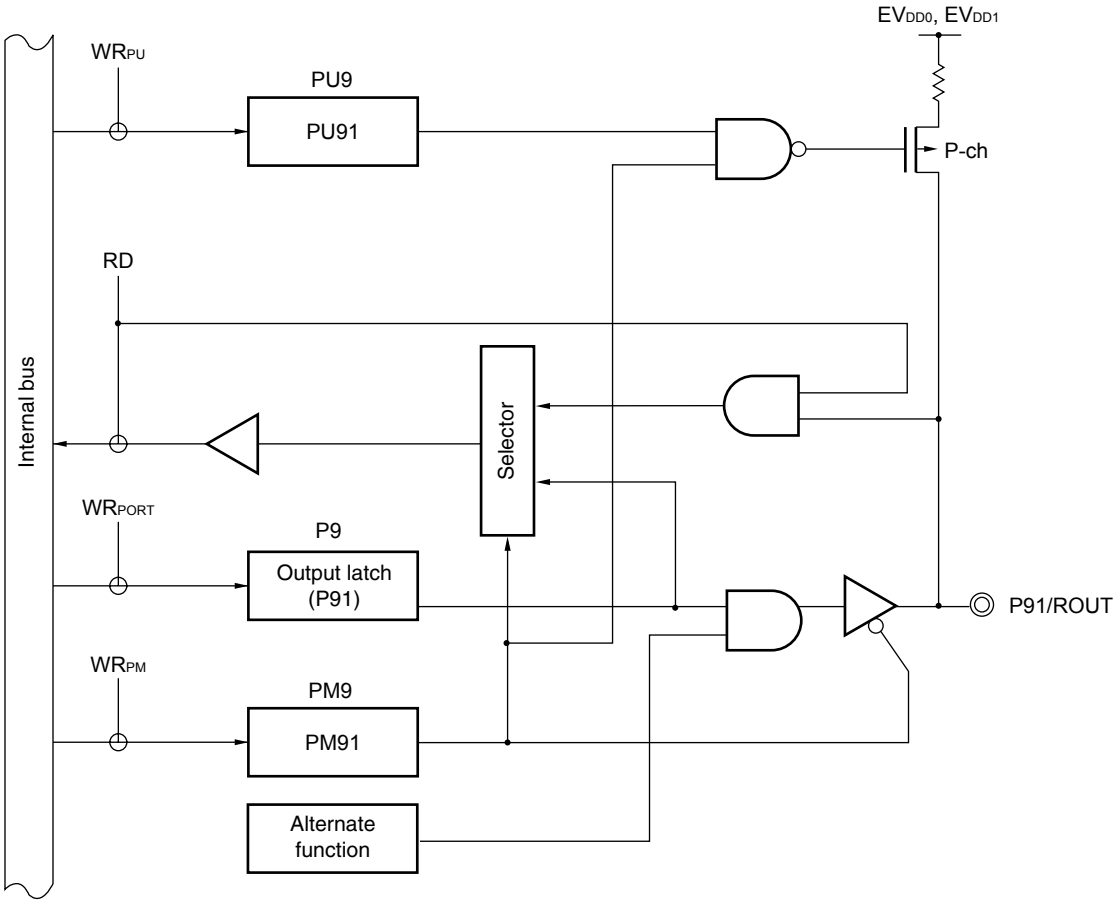
Caution To use P91/ROUT as a general-purpose port, set the remote controller receive data slew control register (RMSW) to "00", which is the same as their default status settings.

Figure 4-36. Block Diagram of P90



P9: Port register 9
PU9: Pull-up resistor option register 9
PM9: Port mode register 9
RD: Read signal
WR××: Write signal

Figure 4-37. Block Diagram of P91



P9: Port register 9
PU9: Pull-up resistor option register 9
PM9: Port mode register 9
RD: Read signal
WR××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

4.2.11 Port 11

Port 11 is a 2-bit I/O port with an output latch. Port 11 can be set to the input mode or output mode in 1-bit units using port mode register 11 (PM11). When the P110 and P111 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 11 (PU11).

This port can also be used for serial data I/O for CEC.

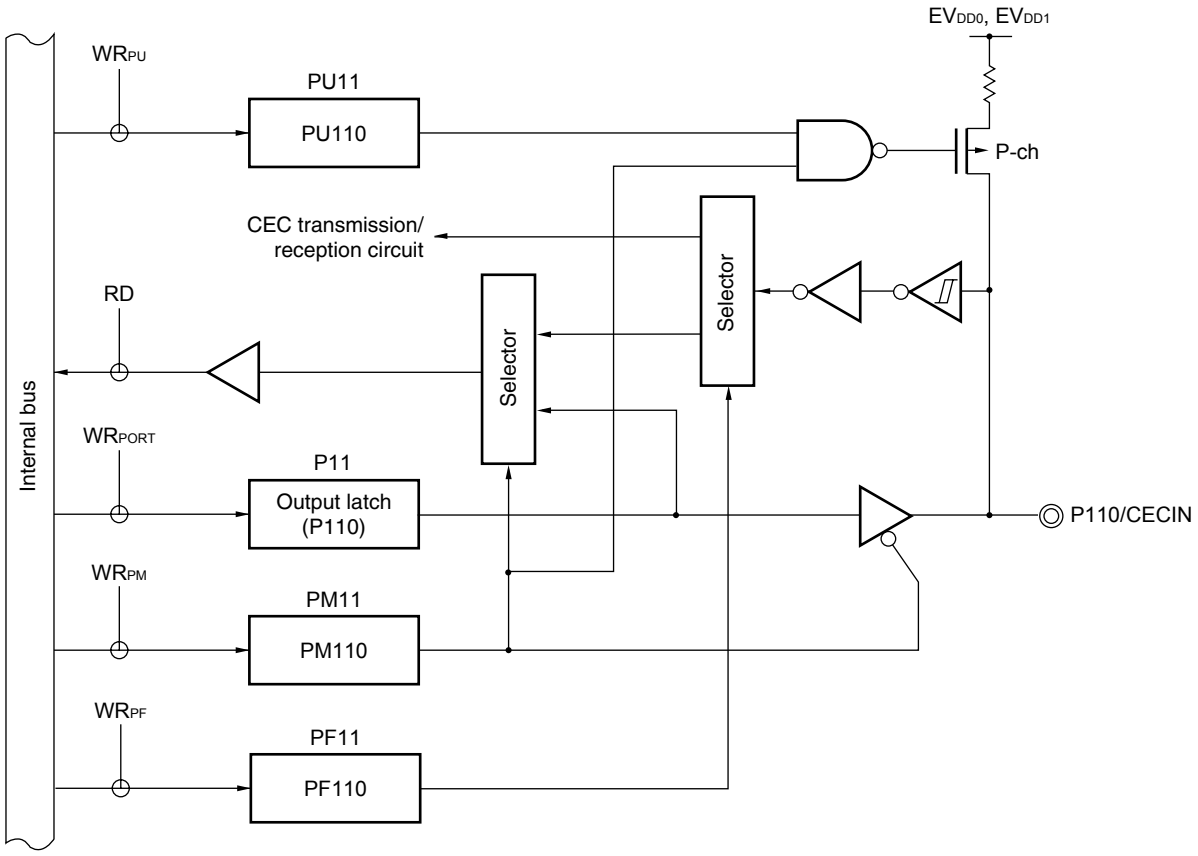
The I/O port mode or CECIN mode can be specified for the P110 pin by using port function register 11 (PF11) (see **Figure 4-60**).

The I/O port mode or CECOUT mode can be specified for the P111 pin by using port function register 11 (PF11) (see **Figure 4-60**).

Reset signal generation sets port 11 to input mode.

Figures 4-38 and 4-39 show block diagrams of port 11.

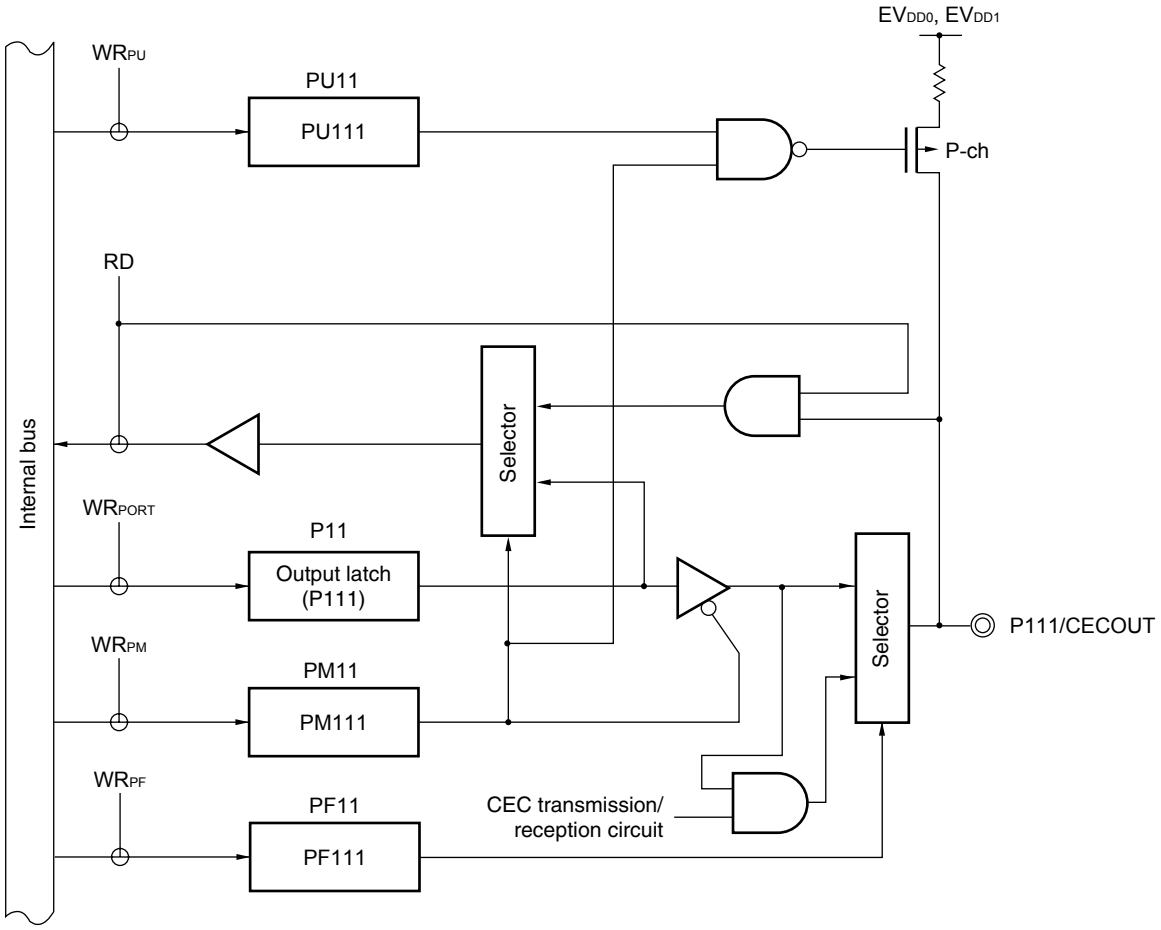
Figure 4-38. Block Diagram of P110



P11: Port register 11
PU11: Pull-up resistor option register 11
PM11: Port mode register 11
PF11: Port function register 11
RD: Read signal
WR××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Figure 4-39. Block Diagram of P111



P11: Port register 11
PU11: Pull-up resistor option register 11
PM11: Port mode register 11
PF11: Port function register 11
RD: Read signal
WR××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

4.2.12 Port 12

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

P121 to P124 are 4-bit input ports.

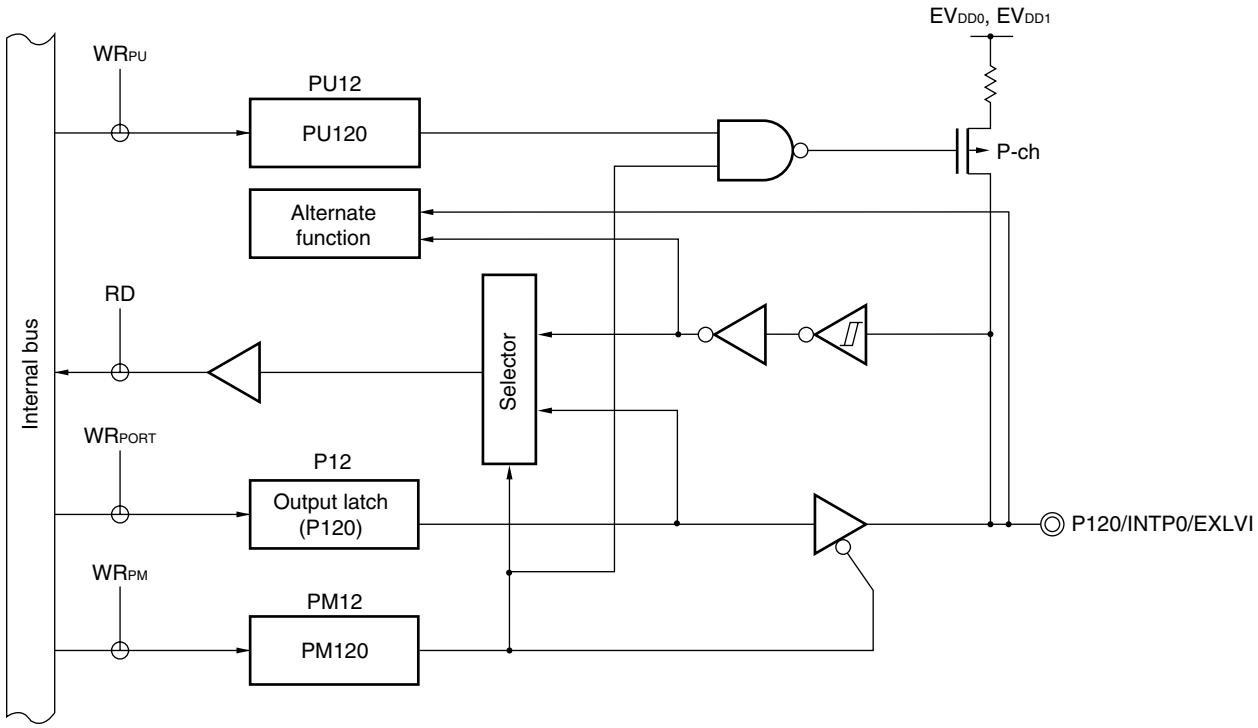
This port can also be used for external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, and external clock input for main system clock.

Reset signal generation sets port 12 to input mode.

Figures 4-40 to 4-42 show block diagrams of port 12.

Caution 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.

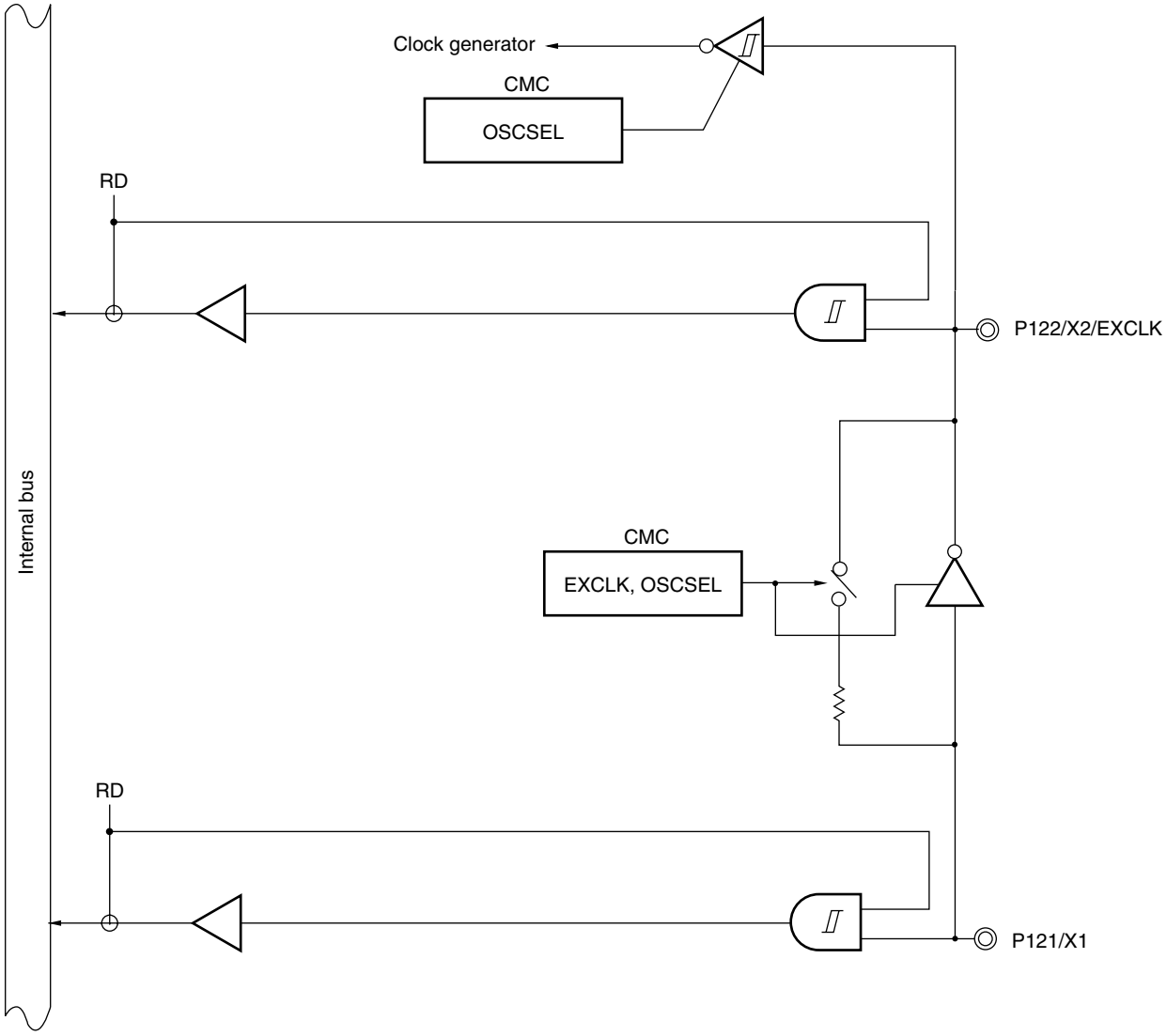
Figure 4-40. Block Diagram of P120



P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
RD: Read signal
WR××: Write signal

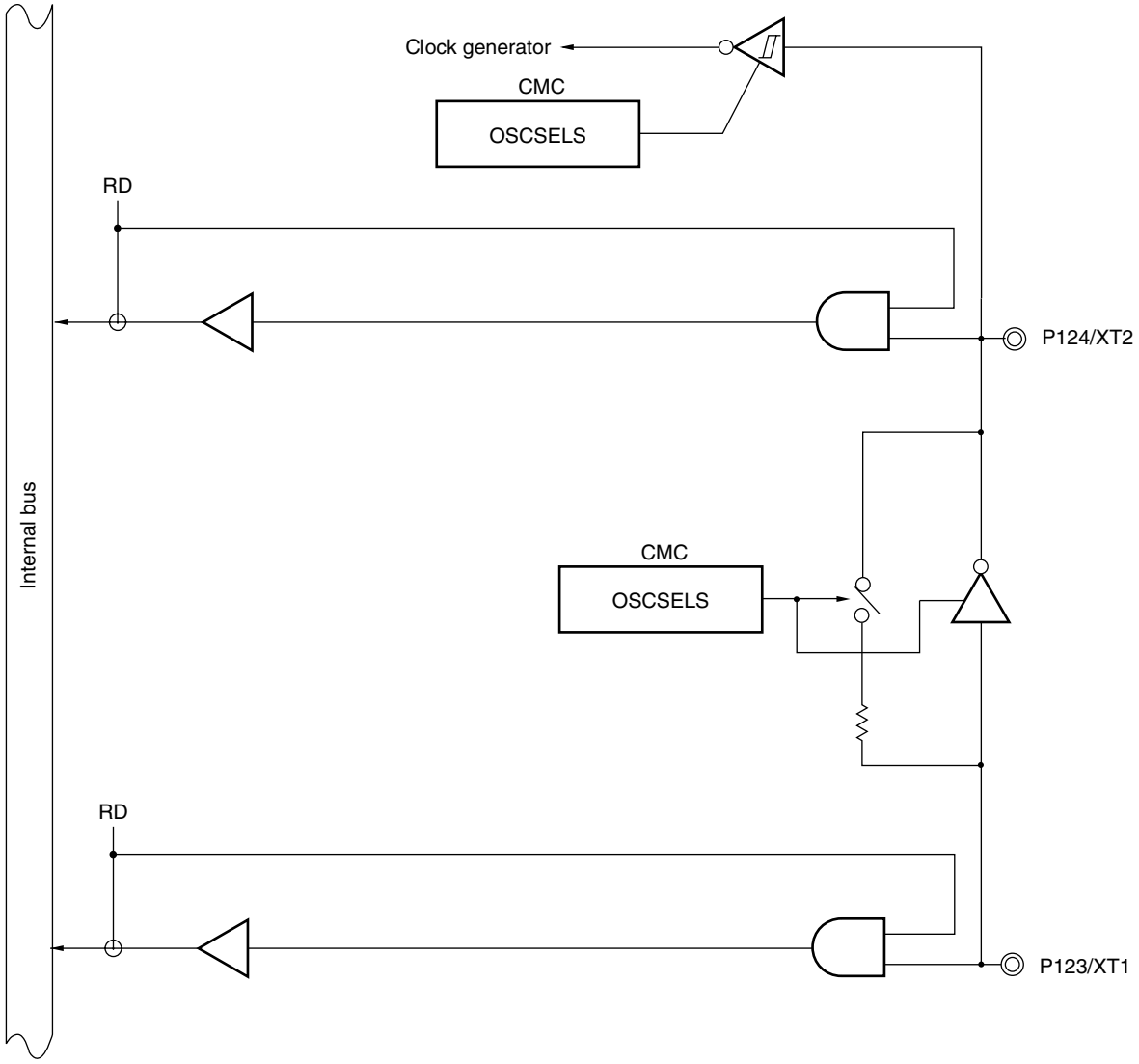
Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Figure 4-41. Block Diagram of P121 and P122



CMC: Clock operation mode control register
RD: Read signal

Figure 4-42. Block Diagram of P123 and P124



CMC: Clock operation mode control register
RD: Read signal

4.2.13 Port 13

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P130	√	√
P131/TI06/TO06	–Note	√

Note TI06/TO06 is shared with following pin, in the 78K0R/KF3-C.
P06/TI06/TO06

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).

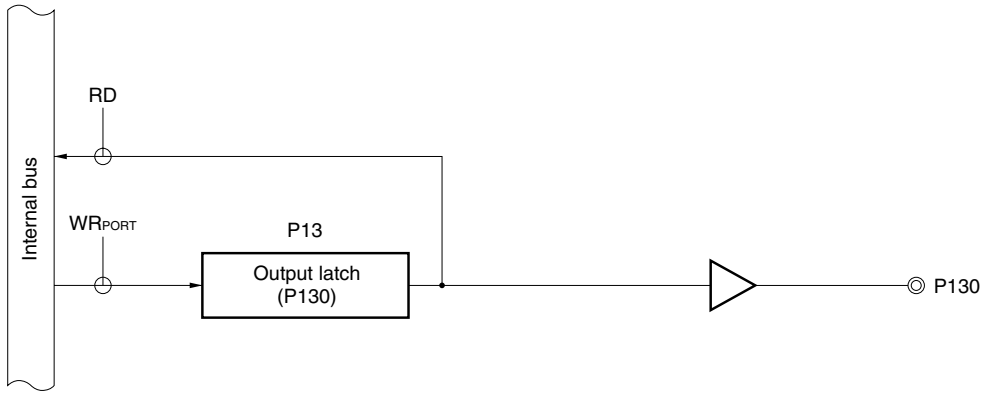
Reset signal generation sets port 13 to input mode.

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××: Write signal

Remark The P130 pin outputs a low level when it is used as a port function pin and a reset is effected. If P130 is set to output a high level, the output signal of P130 can be dummy-output as the CPU reset signal.

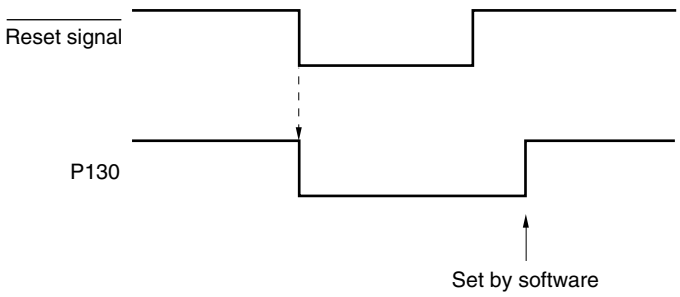
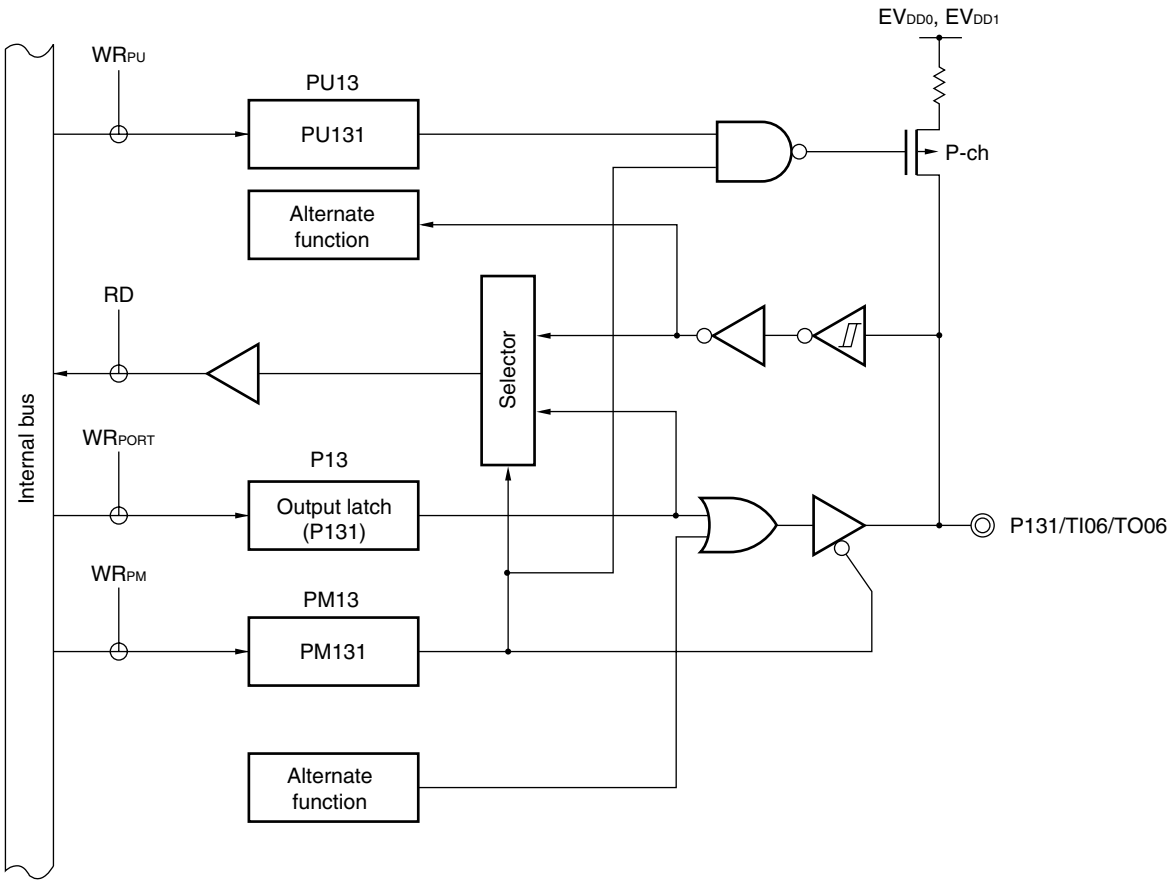


Figure 4-44. Block Diagram of P131



P13: Port register 13
PU13: Pull-up resistor option register 13
PM13: Port mode register 13
RD: Read signal
WR××: Write signal

4.2.14 Port 14

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P140/PCLBUZ0	√	√
P141/PCLBUZ1/INTP7	–Note	√
P142/$\overline{\text{SCK20}}$/SCL20	√	√
P143/SI20/RxD2/SDA20	√	√
P144/SO20/TxD2	√	√
P145/TI07/TO07	–Note	√

Note PCLBUZ1/INTP7 and TI07/TO07 are shared with following pins, respectively, in the 78K0R/KF3-C.
P55/PCLBUZ1/INTP7, P54/TI07/TO07

Port 14 is an I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140 to 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.

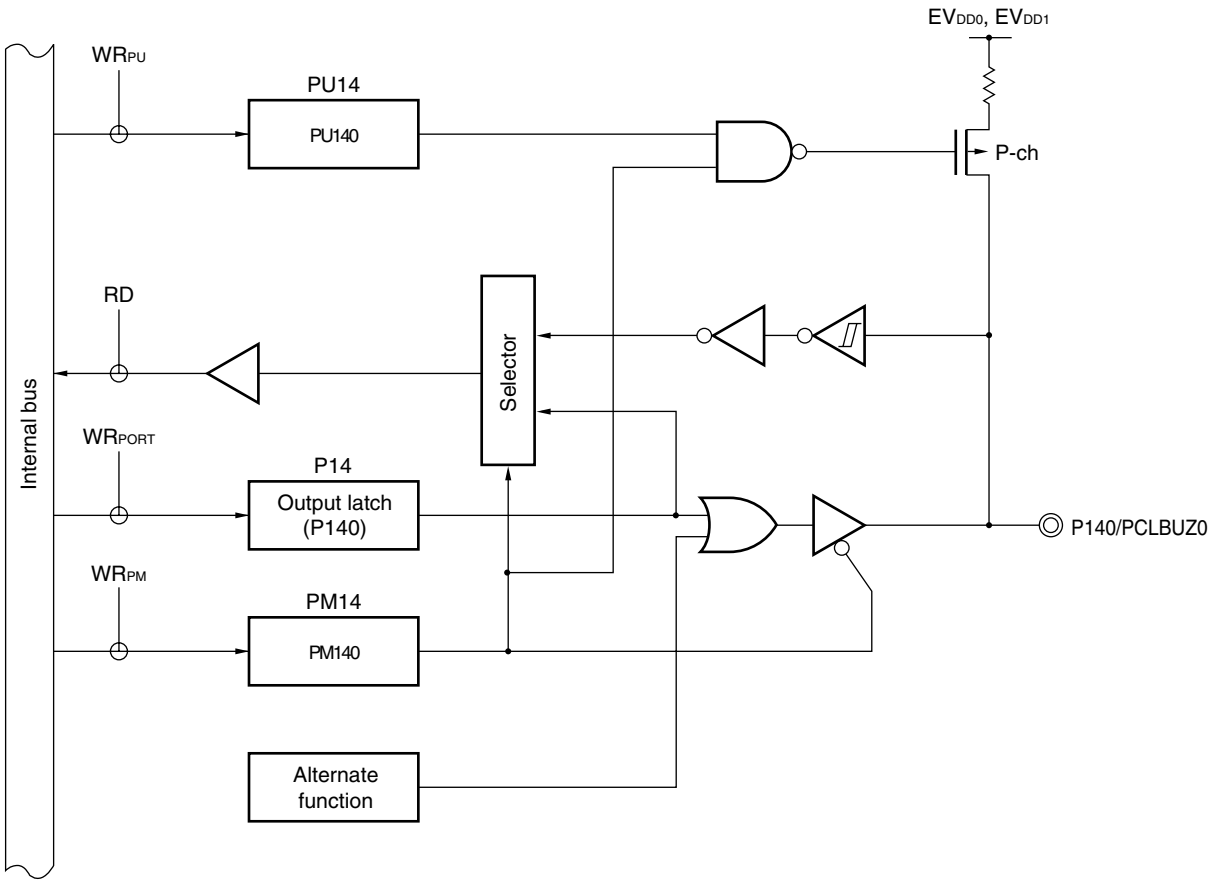
Figures 4-45 to 4-48 show block diagrams of port 14.

Cautions 1. To use P142/$\overline{\text{SCK20}}$/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 11-9 Relationship Between Register Settings and Pins (Channel 0 of Unit 1: CSI20, UART2 Transmission, IIC20)**
- **Table 11-10 Relationship Between Register Settings and Pins (Channel 1 of Unit 1: UART2 Reception)**

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 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.

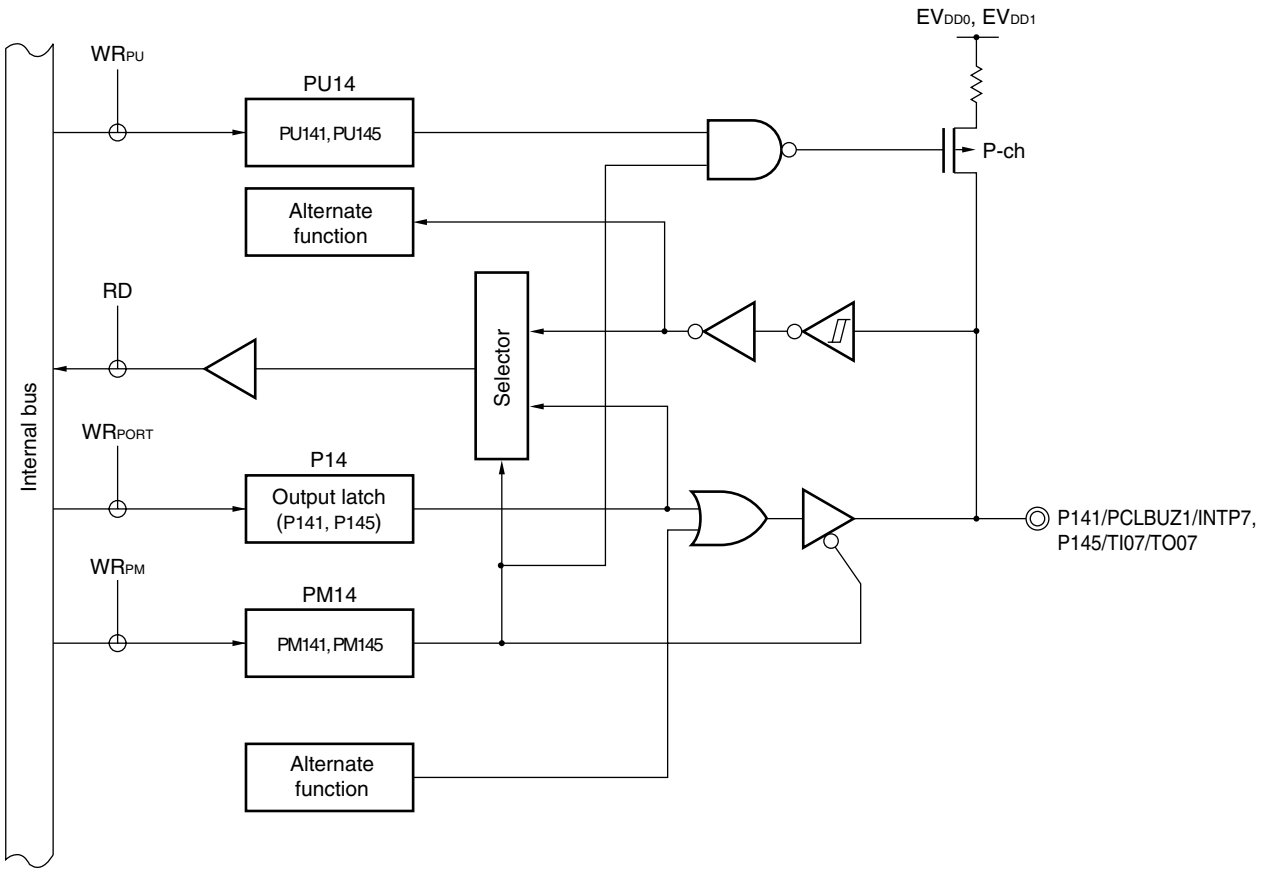
Figure 4-45. Block Diagram of P140



P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
RD: Read signal
WR××: Write signal

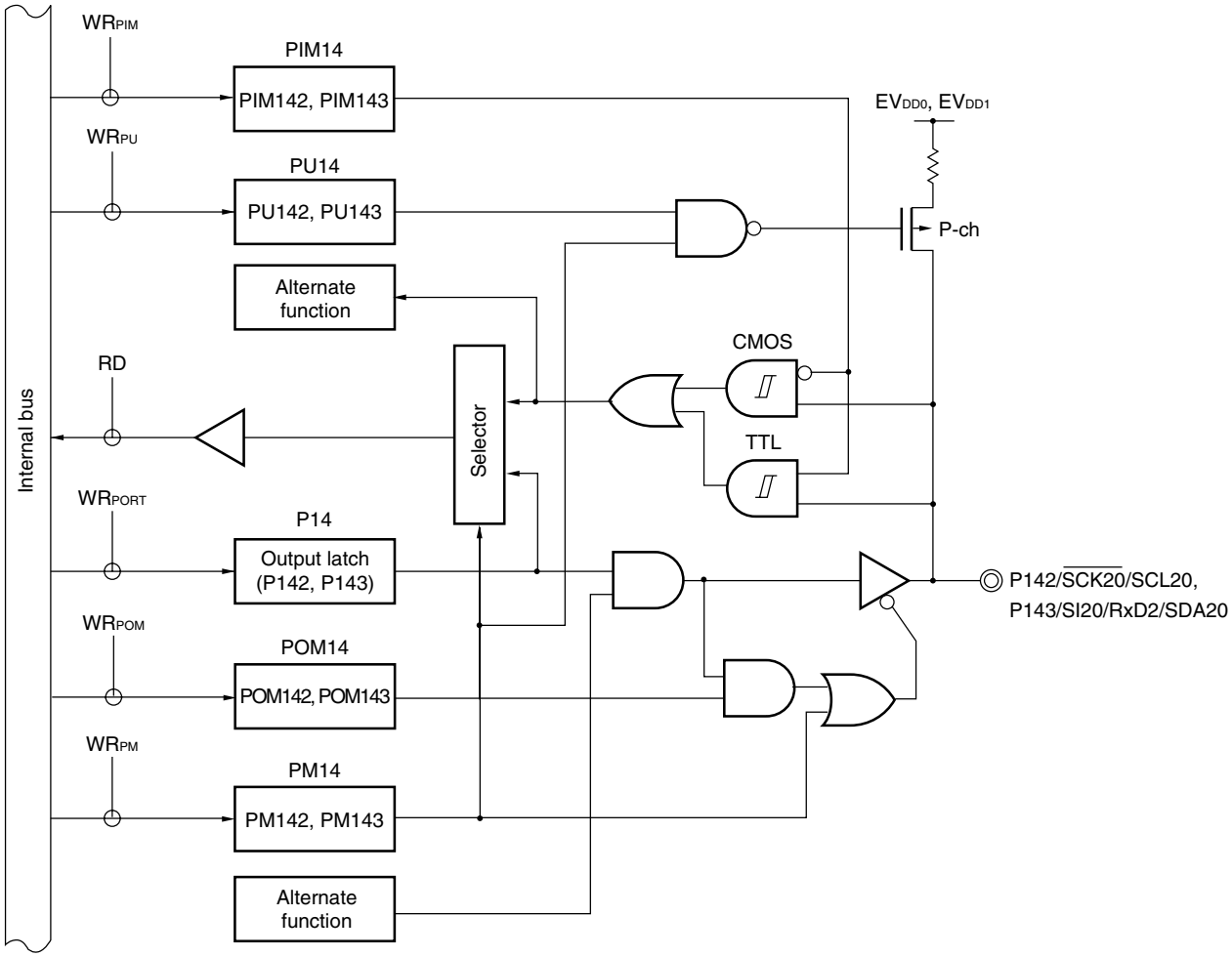
Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Figure 4-46. Block Diagram of P141 and P145



P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
RD: Read signal
WR××: Write signal

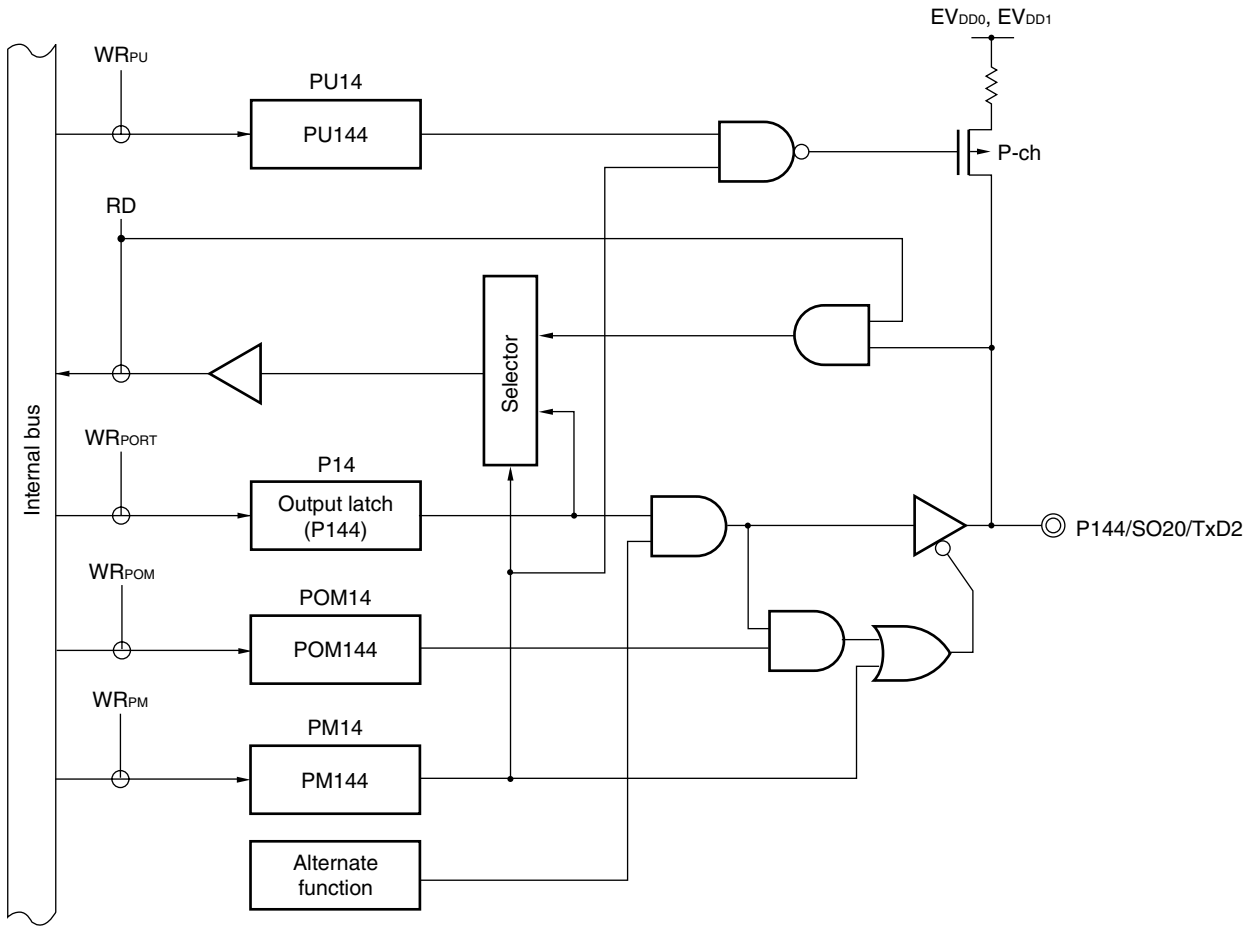
Figure 4-47. Block Diagram of P142 and P143



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××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Figure 4-48. Block Diagram of P144



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××: Write signal

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

4.2.15 Port 15

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
P150/ANI8	√	√
P151/ANI9	√	√
P152/ANI10	√	√
P153/ANI11	√	√
P154/ANI12	–	√
P155/ANI13	–	√
P156/ANI14	–	√
P157/ANI15	–	√

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

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

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 4-6. Setting Functions of P150/ANI8 to P157/ANI15 Pins

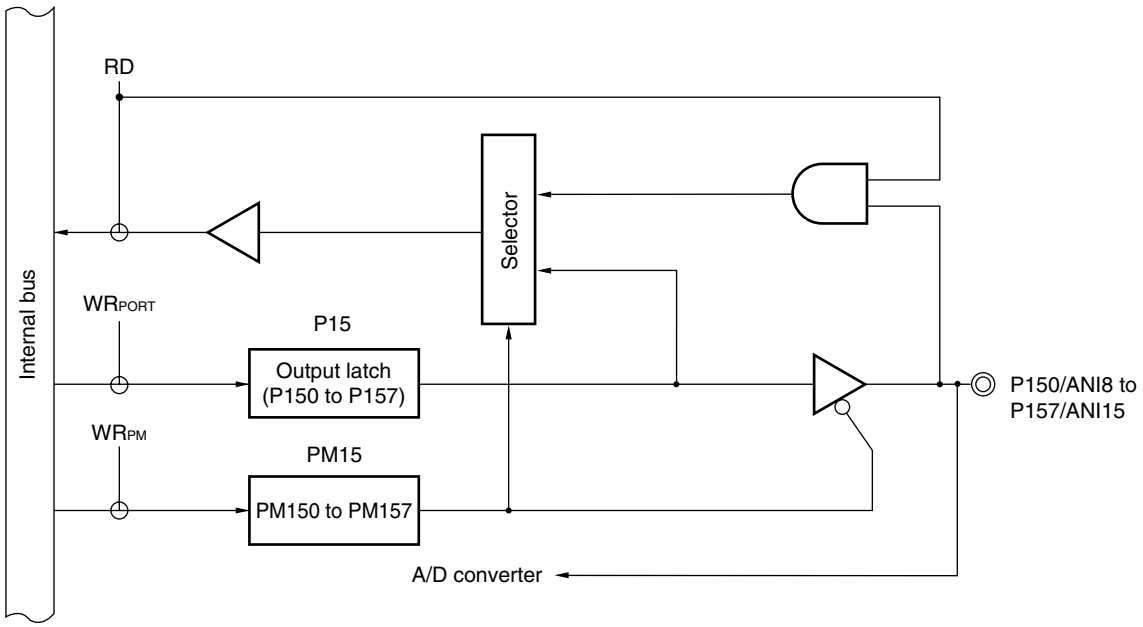
ADPC	PM15	ADS	P150/ANI8 to P157/ANI15 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited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

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

Figure 4-49 shows a block diagram of port 15.

Caution See 2.2.16 AV_{REF} for the voltage to be applied to the AV_{REF} pin when using port 15 as a digital I/O.

Figure 4-49. Block Diagram of P150 to P157



P15: Port register 15
PM15: Port mode register 15
RD: Read signal
WR××: Write signal

4.3 Registers Controlling Port Function

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

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port function register 6 (PF6)
- Port function register 11 (PF11)
- Port input mode registers (PIM0, PIM1, PIM6, PIM14)
- Port output mode registers (POM0, POM1, POM14)
- A/D port configuration register (ADPC)

(1) Port mode registers (Pxx)

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 PF11, PF6, Port Mode Register, and Output Latch When Using Alternate Function**.

Figure 4-50. Format of Port Mode Register (78K0R/KF3-C)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	PM06	PM05	PM04	PM03	PM02	1	1	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	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	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	1	1	1	1	1	1	PM91	PM90	FFF29H	FFH	R/W
PM11	1	1	1	1	1	1	PM111	PM110	FFF2BH	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	1	1	1	PM144	PM143	PM142	1	PM140	FFF2EH	FFH	R/W
PM15	1	1	1	1	PM153	PM152	PM151	PM150	FFF2FH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 7, 9, 11, 12, 14, 15; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bits 0, 1, and 7 of PM0, bits 2 to 7 of PM3, bits 6 and 7 of PM5, bits 2 to 7 of PM9, bits 2 to 7 of PM11, bits 1 to 7 of PM12, bits 1 and 5 to 7 of PM14, and bits 4 to 7 of PM15 to "1".

Figure 4-51. Format of Port Mode Register (78K0R/KG3-C)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	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	1	1	1	1	1	1	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	1	1	1	1	1	1	PM91	1	FFF29H	FFH	R/W
PM11	1	1	1	1	1	1	PM111	PM110	FFF2BH	FFH	R/W
PM12	1	1	1	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

PMmn	Pmn pin I/O mode selection (m = 0 to 9, 11 to 15; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Caution Be sure to set bit 7 of PM0, bits 2 to 7 of PM3, bits 0 and 2 to 7 of PM9, bits 2 to 7 of PM11, bits 1 to 7 of PM12, bits 2 to 7 of PM13, and bits 6 and 7 of PM14 to "1". And be sure to set bit 0 of PM13 to "0".

(2) Port registers (Pxx)

These registers set the output latch value of a port.

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[Note].

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

Reset signal generation clears these registers to 00H.

Note 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.

Figure 4-52. Format of Port Register (78K0R/KF3-C)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	P06	P05	P04	P03	P02	0	0	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	0	0	0	0	0	0	P31	P30	FFF03H	00H (output latch)	R/W
P4	P47	P46	P45	P44	P43	P42	P41	P40	FFF04H	00H (output latch)	R/W
P5	0	0	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
P9	0	0	0	0	0	0	P91	P90	FFF09H	00H (output latch)	R/W
P11	0	0	0	0	0	0	P111	P110	FFF0BH	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W[Note]
P13	0	0	0	0	0	0	0	P130	FFF0DH	00H (output latch)	R/W
P14	0	0	0	P144	P143	P142	0	P140	FFF0EH	00H (output latch)	R/W
P15	0	0	0	0	P153	P152	P151	P150	FFF0FH	00H (output latch)	R/W

Pmn	m = 0 to 7, 9, 11 to 15; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note P121 to P124 are read-only.

Figure 4-53. Format of Port Register (78K0R/KG3-C)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	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	0	0	0	0	0	0	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	0	0	0	0	0	0	P91	0	FFF09H	00H (output latch)	R/W
P11	0	0	0	0	0	0	P111	P110	FFF0BH	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W[Note]
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

Pmn	m = 0 to 9, 11 to 15; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note P121 to P124 are read-only.

(3) Pull-up resistor option registers (PUxx)

These registers specify whether the on-chip pull-up resistors are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode of the pins to which the use of an on-chip pull-up resistor has been specified. 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 pull-up resistor option register.

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 Pull-up Resistor Option Register (78K0R/KF3-C)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	PU06	PU05	PU04	PU03	PU02	0	0	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	F0034H	00H	R/W
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU6	PU67	PU66	PU65	PU64	0	PU62	0	0	F0036H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU9	0	0	0	0	0	0	PU91	PU90	F0039H	00H	R/W
PU11	0	0	0	0	0	0	PU111	PU110	F003BH	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	0	0	0	PU144	PU143	PU142	0	PU140	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 7, 9, 11, 12, 14; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

<R>

Figure 4-55. Relationship Between PF6 Register and PU6 Register

PF62	PU62	Diode connection
0	0	Does not connect pull-up resistor to diode
0	1	
1	0	
1	1	Connects pull-up resistor to diode

Caution CECIO and CECIN/CECOUT pins must not be used at the same time.
Do not set PF110 and PF111 to 1 while PF62 = 1.
Do not set PF62 to 1 while PF110 and PF111 = 1.

Figure 4-56. Format of Pull-up Resistor Option Register (78K0R/KG3-C)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	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	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	F0034H	00H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU6	PU67	PU66	PU65	PU64	0	PU62	0	0	F0036H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU8	PU87	PU86	PU85	PU84	PU83	PU82	PU81	PU80	F0038H	00H	R/W
PU9	0	0	0	0	0	0	PU91	0	F0039H	00H	R/W
PU11	0	0	0	0	0	0	PU111	PU110	F003BH	00H	R/W
PU12	0	0	0	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

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 9, 11 to 14; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

<R>

Figure 4-57. Relationship Between PF6 Register and PU6 Register

PF62	PU62	Diode connection
0	0	Does not connect pull-up resistor to diode
0	1	
1	0	
1	1	Connects pull-up resistor to diode

Caution CECIO and CECIN/CECOUT pins must not be used at the same time.
Do not set PF110 and PF111 to 1 while PF62 = 1.
Do not set PF62 to 1 while PF110 and PF111 = 1.

(4) Port function register 6 (PF6)

This register sets whether to use pin P62 as I/O port mode or CECIO mode.

Input to the P62 pin can be specified through a normal input buffer or a CEC input buffer, using port function register 6 (PF6) and port input mode register 6 (PIM6).

Whether to connect a diode can be set by setting the PF6 register and pull-up resistor option register 6 (PU6).

Figure 4-58. Format of Port Function Register 6 (PF6)

Address: F0076H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF6	0	0	0	0	0	PF62	0	0

PF62	P62 operation mode specification
0	Used as I/O port mode
1	Used as CECIO mode

Figure 4-59. Relationship Between PF6 Register, PIM6 Register, and PU6 Register

PF62	PIM62	Input buffer specification
0	0	Normal input buffer
0	1	
1	0	
1	1	CEC input buffer

PF62	PU62	Diode connection
0	0	Does not connect pull-up resistor to diode
0	1	
1	0	
1	1	Connects pull-up resistor to diode

Caution CECIO and CECIN/CECOUT pins must not be used at the same time.
Do not set PF110 and PF111 to 1 while PF62 = 1.
Do not set PF62 to 1 while PF110 and PF111 = 1.

(5) Port function register 11 (PF11)

This register sets whether to use pin P110/CECIN as I/O port mode or CECIN mode.

This register sets whether to use pin P111/CECOUT as I/O port mode or CECOUT mode.

Figure 4-60. Format of Port Function Register 11 (PF11)

Address: F007BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PF11	0	0	0	0	0	0	PF111	PF110

PF110	P110 operation mode specification
0	Used as I/O port mode
1	Used as CECIN mode

PF111	P111 operation mode specification
0	Used as I/O port mode
1	Used as CECOUT mode

Caution CECIO and CECIN/CECOUT pins must not be used at the same time.

Do not set PF110 and PF111 to 1 while PF62 = 1.

Do not set PF62 to 1 while PF110 and PF111 = 1.

(6) Port input mode registers (PIM0, PIM1, PIM6, PIM14)

These registers set the input buffer of P03, P04, P10, P11, or P62 in 1-bit units.

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

Input to the P62 pin can be specified through a normal input buffer or a CEC input buffer, using port input mode register 6 (PIM6) and port function register 6 (PF6).

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

Reset signal generation clears these registers to 00H.

Figure 4-61. 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
PIM1	0	0	0	0	0	0	PIM11	PIM10	F0044H	00H	R/W
PIM6	0	0	0	0	0	PIM62	0	0	F0046H	00H	R/W
PIM14	0	0	0	0	PIM143	PIM142	0	0	F004EH	00H	R/W

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

Figure 4-62. Relationship Between PIM6 Register and PF6 Register

PIM62	PF62	Input buffer specification
0	0	Normal input buffer
0	1	
1	0	
1	1	CEC input buffer

(7) Port output mode registers (POM0, POM1, POM14)

These registers set the output mode of P02 to P04, P10, P12, 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 and SDA20 pins during simplified I^2C 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 4-63. 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
POM1	0	0	0	0	0	POM12	0	POM10	F0051H	00H	R/W
POM14	0	0	0	POM144	POM143	POM142	0	0	F005EH	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 1, 14; n = 0, 2 to 4)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

(8) 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 4-64. 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	A
0	0	0	0	1	A	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	A	D	D
0	0	0	1	1	A	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	A	D	D	D	D
0	0	1	0	1	A	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	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	D
0	1	0	0	0	A	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	D
0	1	0	1	0	A	A	A	A	A	A	D	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	D
0	1	1	0	0	A	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	D
0	1	1	1	0	A	A	D	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	D
1	0	0	0	0	D	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. Be sure to first set the ADCEN bit of peripheral enable register 0 (PER0) to 1 when setting up the ADPC register. If ADCEN = 0, writing to the ADPC register is ignored and specified values are returned to the initial values.

Remark P20/ANI0 to P27/ANI7, P150/ANI8 to P153/ANI11: 78K0R/KF3-C
P20/ANI0 to P27/ANI7, P150/ANI8 to P157/ANI15: 78K0R/KG3-C

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, 1, 6, and 14 operate with V_{DD} = 4.0 V to 5.5 V, I/O connections with an external device that operates on 2.5 V, 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, PIM1, PIM6, 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, POM1, POM14).

(1) Setting procedure when using I/O pins of UART0, UART1, UART2, CSI00, CSI10, and CSI20 functions

(a) Use as 2.5 V, 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 UART0:	P11
In case of UART1:	P03
In case of UART2:	P143
In case of CSI00:	P10, P11
In case of CSI10:	P03, P04
In case of CSI20:	P142, P143

<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, 3 V operating voltage.

(b) Use as 2.5 V, 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:	P12
In case of UART1:	P02
In case of UART2:	P144
In case of CSI00:	P10, P12
In case of CSI10:	P02, P04
In case of CSI20:	P142, P144

<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> Communication is started by setting the serial array unit.

Remark n = 0, 1, 14

(2) Setting procedure when using I/O pins of simplified IIC10 and IIC20 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 IIC20: P142, P143

<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 IIC mode.

Remark n = 0, 14

4.5 Settings of PF11, PF6, Port Mode Register, and Output Latch When Using Alternate Function

To use the alternate function of a port pin, set the PF11, PF6, port mode register, and output latch as shown in Table 4-7.

Table 4-7. Settings of PF11, PF6, Port Mode Register, and Output Latch When Using Alternate Function (1/4)

Pin Name	Alternate Function		PF11	PF6	PM××	P××
	Function Name	I/O				
P00	TI00Note	Input			1	×
P01	TO00Note	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	$\overline{\text{SCK10}}$	Input			1	×
		Output			0	1
	SCL10	I/O			0	1
P05	TI05Note	Input			1	×
	TO05Note	Output			0	0
P06	TI06Note	Input			1	×
	TO06Note	Output			0	0
P10	$\overline{\text{SCK00}}$	Input			1	×
		Output			0	1
P11	SI00	Input			1	×
	RxD0	Input			1	×
P12	SO00	Output			0	1
	TxD0	Output			0	1
P15	RTCDIV	Output			0	0
	RTCCL	Output			0	0
P16	TI01	Input			1	×
	TO01	Output			0	0
	INTP5	Input			1	×
P17	TI02	Input			1	×
	TO02	Output			0	0

Remark ×: don't care
PM××: Port mode register
P××: Port output latch

Note The ports with which TI00, TO00, TI05/TO05, TI06/TO06, and TI07/TO07 pins are shared differ depending on the product.
78K0R/KF3-C: P53/TI00, P52/TO00, P05/TI05/TO05, P06/TI06/TO06, P54/TI07/TO07
78K0R/KG3-C: P00/TI00, P01/TO00, P46/INTP1/TI05/TO05/RIN01, P131/TI06/TO06, P145/TI07/TO07

Table 4-7. Settings of PF11, PF6, Port Mode Register, and Output Latch When Using Alternate Function (2/4)

Pin Name	Alternate Function		PF11	PF6	PM××	P××
	Function Name	I/O				
P20 to P27[Note 1]	ANI0 to ANI7[Note 1]	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	$\overline{\text{SCK01}}$	Input			1	×
		Output			0	1
P44	SI01	Input			1	×
P45	SO01	Output			0	1
P46	TI05[Note 2]	Input			1	×
	TO05[Note 2]	Output			0	0
	INTP1[Note 3]	Input			1	×
	RIN01	Input			1	×

Remark ×: don't care
PM××: Port mode register
P××: Port output latch

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.

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.	

In addition, the analog input pins mounted depend on the product.
P20/ANI0 to P27/ANI7, P150/ANI8 to P153/ANI11: 78K0R/KF3-C
P20/ANI0 to P27/ANI7, P150/ANI8 to P157/ANI15: 78K0R/KG3-C

2. The ports with which TI00, TO00, TI05/TO05, TI06/TO06, and TI07/TO07 pins are shared differ depending on the product.
78K0R/KF3-C: P53/TI00, P52/TO00, P05/TI05/TO05, P06/TI06/TO06, P54/TI07/TO07
78K0R/KG3-C: P00/TI00, P01/TO00, P46/INTP1/TI05/TO05/RIN01, P131/TI06/TO06, P145/TI07/TO07

3. The ports with which PCLBUZ1/INTP7, INTP1, and INTP2 pins are shared differ depending on the product.
78K0R/KF3-C: P55/PCLBUZ1/INTP7, P50/INTP1, P51/INTP2
78K0R/KG3-C: P141/PCLBUZ1/INTP7, P46/INTP1/TI05/TO05/RIN01, P47/INTP2

Table 4-7. Settings of PF11, PF6, Port Mode Register, and Output Latch When Using Alternate Function (3/4)

Pin Name	Alternate Function		PF11	PF6	PM××	P××
	Function Name	I/O				
P47	INTP2[Note 1]	Input			1	×
	RIN23	Input			1	×
P50	INTP1[Note 1]	Input			1	×
P51	INTP2[Note 1]	Input			1	×
P52	TO00[Note 2]	Output			0	0
P53	TI00[Note 2]	Input			1	×
P54	TI07[Note 2]	Input			1	×
	TO07[Note 2]	Output			0	0
P55	PCLBUZ1[Note 1]	Output			0	0
	INTP7[Note 1]	Input			1	×
P60	SCL0	I/O			0	0
P61	SDA0	I/O			0	0
P62	CECIO	I/O		PF62 = 1	0	1
P64	TI10	Input			1	×
	TO10	Output			0	0
P65	TI11	Input			1	×
	TO11	Output			0	0
P66	TI12	Input			1	×
	TO12	Output			0	0
P70 to P73	KR0 to KR3	Input			1	×
P74	INTP8	Input			1	×
	KR4	Input			1	×
P75 to P77	KR5 to KR7	Input			1	×
P91	ROUT	Output			0	1
P110	CECIN	Input	PF110 = 1		1	×
P111	CECOUT	Output	PF111 = 1		0	1
P120	INTP0	Input			1	×
	EXLVI	Input			1	×

Remark ×: don't care
PM××: Port mode register
P××: Port output latch

Notes 1. The ports with which PCLBUZ1/INTP7, INTP1, and INTP2 pins are shared differ depending on the product.
78K0R/KF3-C: P55/PCLBUZ1/INTP7, P50/INTP1, P51/INTP2
78K0R/KG3-C: P141/PCLBUZ1/INTP7, P46/INTP1/TI05/TO05/RIN01, P47/INTP2

2. The ports with which TI00, TO00, TI05/TO05, TI06/TO06, and TI07/TO07 pins are shared differ depending on the product.
78K0R/KF3-C: P53/TI00, P52/TO00, P05/TI05/TO05, P06/TI06/TO06, P54/TI07/TO07
78K0R/KG3-C: P00/TI00, P01/TO00, P46/INTP1/TI05/TO05/RIN01, P131/TI06/TO06, P145/TI07/TO07

Table 4-7. Settings of PF11, PF6, Port Mode Register, and Output Latch When Using Alternate Function (4/4)

Pin Name	Alternate Function		PF11	PF6	PM××	P××
	Function Name	I/O				
P131	TI06Note 1	Input			1	×
	TO06Note 1	Output			0	0
P140	PCLBUZ0	Output			0	0
P141	PCLBUZ1Note 2	Output			0	0
	INTP7Note 2	Input			1	×
P142	$\overline{SCK20}$	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	TI07Note 1	Input			1	×
	TO07Note 1	Output			0	0
P150 to P157Note 3	ANI8 to ANI15Note 3	Input			1	×

Remark ×: don't care
PM××: Port mode register
P××: Port output latch

Notes 1. The ports with which TI00, TO00, TI05/TO05, TI06/TO06, and TI07/TO07 pins are shared differ depending on the product.
78K0R/KF3-C: P53/TI00, P52/TO00, P05/TI05/TO05, P06/TI06/TO06, P54/TI07/TO07
78K0R/KG3-C: P00/TI00, P01/TO00, P46/INTP1/TI05/TO05/RIN01, P131/TI06/TO06, P145/TI07/TO07

2. The ports with which PCLBUZ1/INTP7, INTP1, and INTP2 pins are shared differ depending on the product.
78K0R/KF3-C: P55/PCLBUZ1/INTP7, P50/INTP1, P51/INTP2
78K0R/KG3-C: P141/PCLBUZ1/INTP7, P46/INTP1/TI05/TO05/RIN01, P47/INTP2

3. 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.

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.	

In addition, the analog input pins mounted depend on the product.
P20/ANI0 to P27/ANI7, P150/ANI8 to P153/ANI11: 78K0R/KF3-C
P20/ANI0 to P27/ANI7, P150/ANI8 to P157/ANI15: 78K0R/KG3-C

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/Kx3-C.

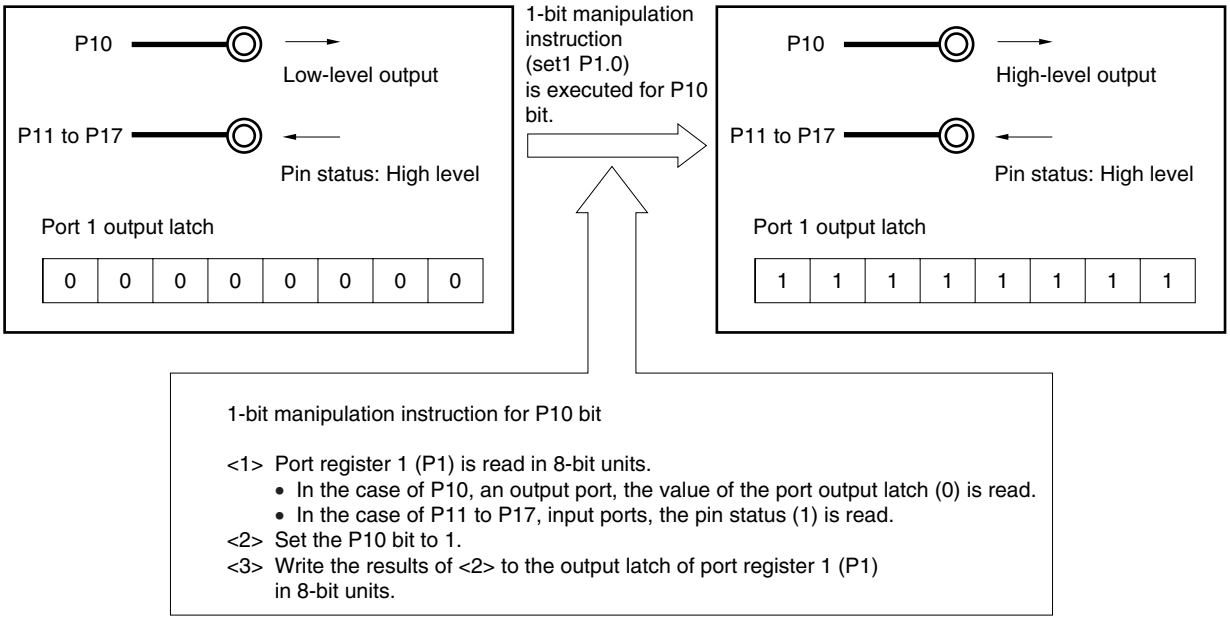
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

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 4-65. Bit Manipulation Instruction (P10)



CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

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

(1) Main system clock

<1> X1 oscillator

This circuit oscillates a clock of f_X = 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[Note]

This circuit oscillates clocks 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 HIOSTOP (bit 0 of CSC).

<3> 20 MHz internal high-speed oscillation clock oscillator[Note]

This circuit oscillates a clock of f_{IH20} = 20 MHz (TYP.). Oscillation can be started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1. Oscillation can be stopped by setting DSCON to 0.

Note To use the 8 or 20 MHz internal high-speed oscillation clock, use the option byte to set the frequency in advance (for details, see **CHAPTER 24 OPTION BYTE**). Also, the internal high-speed oscillator automatically starts oscillating after reset release. To use the 20 MHz internal high-speed oscillator to operate the microcontroller, oscillation is started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1.

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).

Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{IH20}: 20 MHz internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency
f_{SUB}: Subsystem clock frequency

(3) Internal low-speed oscillation clock (clock dedicated to watchdog timer)

- **Internal low-speed oscillator**

  This circuit oscillates a clock of f_{IL} = 30 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_{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

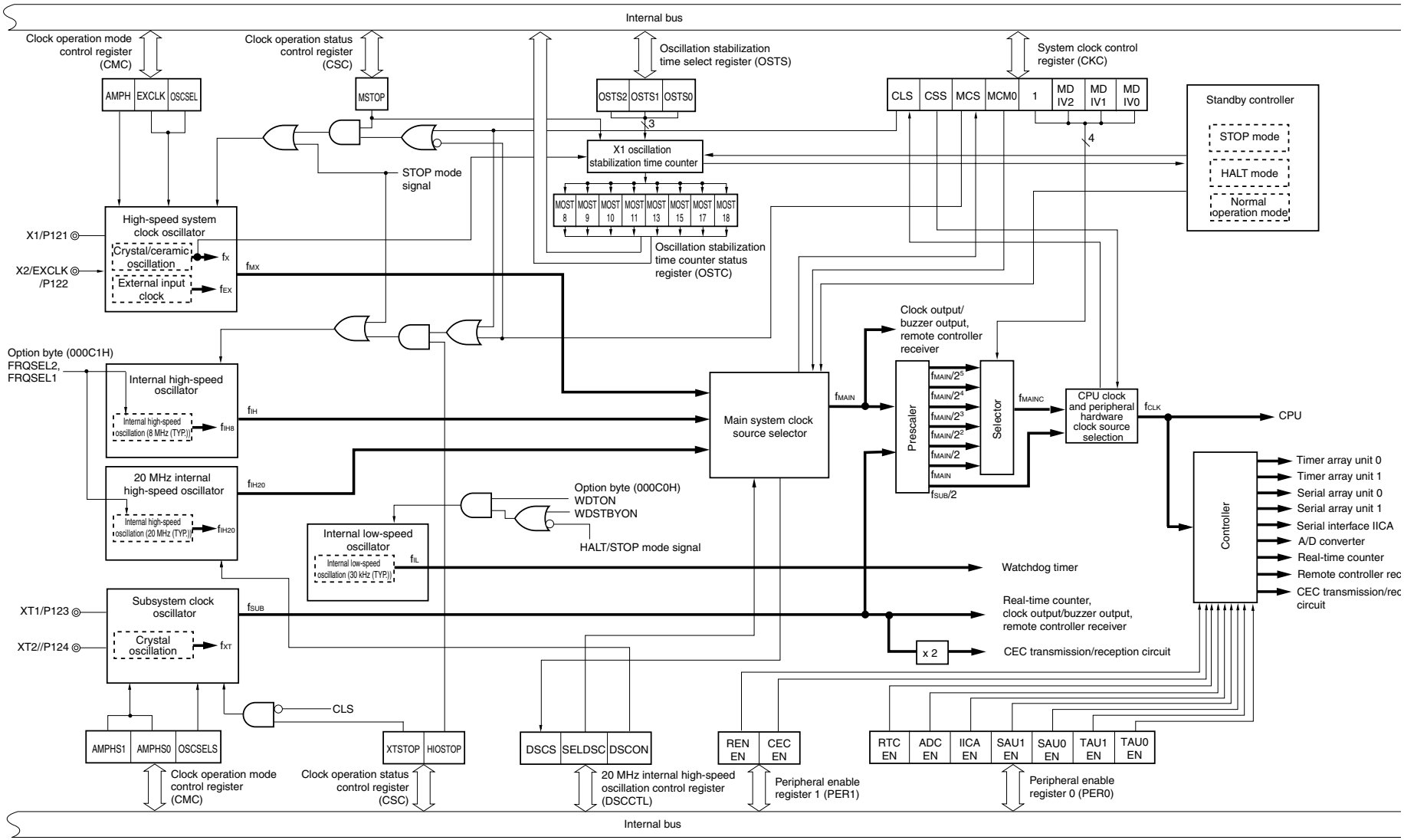
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) System clock control register (CKC) 20 MHz internal high-speed oscillation control register (DSCCTL) Peripheral enable registers 0, 1 (PER0, PER1) Operation speed mode control register (OSMC)
Oscillators	X1 oscillator XT1 oscillator Internal high-speed oscillator Internal low-speed oscillator

Figure 5-1. Block Diagram of Clock Generator



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

Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{IH20}: 20 MHz internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{MAINC}: Main system select clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Internal low-speed oscillation clock frequency

5.3 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)
- 20 MHz internal high-speed oscillation control register (DSCCTL)
- Peripheral enable registers 0, 1 (PER0, PER1)
- Operation speed mode control register (OSMC)

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

Address: FFFA0H After reset: 00H R/W

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

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

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	

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

AMPH	Control of X1 clock oscillation frequency
0	2 MHz ≤ fx ≤ 10 MHz
1	10 MHz < fx ≤ 20 MHz

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. When CMC is used at the default value (00H), be sure to set 00H to this register after reset release in order to prevent malfunctioning during a program loop.

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

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **When using the ultra-low power consumption oscillation (AMPHS1 = 1) as the mode of the XT1 oscillator, use the recommended resonators described in CHAPTER 29 ELECTRICAL SPECIFICATIONS.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1 = 1) is selected.**

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

- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential as V_{SS} as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

Remark f_X: X1 clock 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 20 MHz internal high-speed oscillation clock and 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 5-3. 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	High-speed system clock operation control		
	X1 oscillation mode	External clock input mode	Input port mode
0	X1 oscillator operating	External clock from EXCLK pin is valid	Input port
1	X1 oscillator stopped	External clock from EXCLK pin is invalid	

XTSTOP	Subsystem clock operation control	
	XT1 oscillation mode	Input port mode
0	XT1 oscillator operating	Input port
1	XT1 oscillator stopped	

HIOSTOP	Internal high-speed oscillation clock operation control
0	Internal high-speed oscillator operating
1	Internal high-speed oscillator stopped[Note]

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

Note The 8 MHz (TYP.) internal high-speed oscillation clock stops. Stopping the internal high-speed oscillator (HIOSTOP = 1) is prohibited while the 20 MHz internal high-speed oscillation clock is operating (DSCON = 1). Stop the 20 MHz internal high-speed oscillation clock by using the 20 MHz internal high-speed oscillation control register (DSCCTL) and not the HIOSTOP bit.

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

Table 5-2. Condition Before Stopping Clock Oscillation and Flag Setting

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock External main system clock	CPU and peripheral hardware clocks operate with a clock other than the high-speed system clock. (CLS = 0 and MCS = 0, or CLS = 1)	MSTOP = 1
Subsystem clock	CPU and peripheral hardware clocks operate with a clock other than the subsystem clock. (CLS = 0)	XTSTOP = 1
Internal high-speed oscillation clock	CPU and peripheral hardware clocks operate with a clock other than the internal high-speed oscillator clock and 20 MHz internal high-speed oscillation clock. (CLS = 0 and MCS = 1, or CLS = 1)	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.

Reset signal generation, the STOP instruction, or 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 5-4. 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		
									f_X = 10 MHz	f_X = 20 MHz
0	0	0	0	0	0	0	0	$2^8/f_X$ max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	$2^8/f_X$ min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	$2^9/f_X$ min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	$2^{10}/f_X$ min.	102.4 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	$2^{11}/f_X$ min.	204.8 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	$2^{13}/f_X$ min.	819.2 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	$2^{15}/f_X$ min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	$2^{17}/f_X$ min.	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	$2^{18}/f_X$ min.	26.21 ms min.	13.11 ms min.

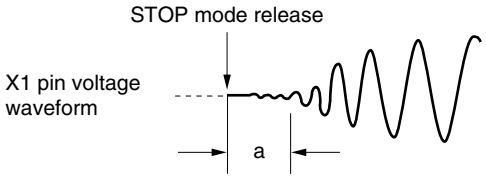
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.

2. 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.)

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

(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. Using OSTC, the oscillation stabilization time up to the time set to OSTS can be checked.

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

Reset signal generation sets OSTS to 07H.

Figure 5-5. 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		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	Setting prohibited
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_X$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_X$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.21 ms	13.11 ms

Cautions 1. To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTS register before executing the STOP instruction.

2. Setting the oscillation stabilization time to 20 μs or less is prohibited.

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

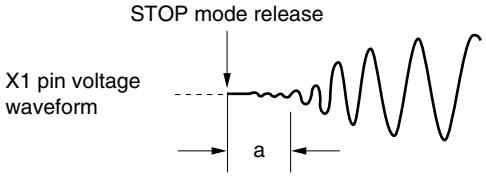
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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

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.)**

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



Remark fx: X1 clock oscillation frequency

(5) 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 5-6. Format of System Clock Control Register (CKC)

Address: FFFA4H After reset: 09H R/W[Note 1]

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_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock divided by 2 ($f_{SUB}/2$)

MCS	Status of Main system clock (f_{MAIN})
0	Internal high-speed oscillation clock (f_{IH}) or 20 MHz internal high-speed oscillation clock (f_{IH20})
1	High-speed system clock (f_{MX})

MCM0	Main system clock (f_{MAIN}) operation control
0	Selects the internal high-speed oscillation clock (f_{IH}) or 20 MHz internal high-speed oscillation clock (f_{IH20}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

CSS	MDIV2	MDIV1	MDIV0	Selection of CPU/peripheral hardware clock (f_{CLK})
0	0	0	0	f_{MAIN}
	0	0	1	$f_{MAIN}/2$ (This is the default setting if MCM0 = 0.)
	0	1	0	$f_{MAIN}/2^2$
	0	1	1	$f_{MAIN}/2^3$
	1	0	0	$f_{MAIN}/2^4$
	1	0	1	$f_{MAIN}/2^5$ [Note 2]
1[Note 3]	×	×	×	$f_{SUB}/2$
Other than above				Setting prohibited

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

2. Setting is prohibited if the high-speed system clock (f_{MX}) is selected as the main system clock (f_{MAIN}) and if $f_{MX} < 4$ MHz.

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

Remarks 1. f_{IH}: Internal high-speed oscillation clock frequency
f_{IH20}: 20 MHz Internal high-speed oscillation clock frequency
f_{MX}: High-speed system clock frequency
f_{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 hardware clock.

3. If the subsystem clock is used as the peripheral hardware clock, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS.

The fastest instruction can be executed in 1 clock of the CPU clock in the 78K0R/Kx3-C. Therefore, the relationship between the CPU clock (f_{CLK}) and the minimum instruction execution time is as shown in Table 5-3.

Table 5-3. 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 10 MHz (TYP.) Operation	At 32.768 kHz Operation
f_{MAIN}	0.1 μs	0.05 μs	0.125 μs (TYP.)	0.05 μs (TYP.)	–
$f_{MAIN}/2$	0.2 μs	0.1 μs	0.25 μs (TYP.) (default)	0.1 μs (TYP.)	–
$f_{MAIN}/2^2$	0.4 μs	0.2 μs	0.5 μs (TYP.)	0.2 μs (TYP.)	–
$f_{MAIN}/2^3$	0.8 μs	0.4 μs	1.0 μs (TYP.)	0.4 μs (TYP.)	–
$f_{MAIN}/2^4$	1.6 μs	0.8 μs	2.0 μs (TYP.)	0.8 μs (TYP.)	–
$f_{MAIN}/2^5$	3.2 μs	1.6 μs	4.0 μs (TYP.)	1.6 μs (TYP.)	–
$f_{SUB}/2$	–		–		61 μs

Remark f_{MAIN}: Main system clock frequency (f_{IH} or f_{MX})
f_{SUB}: Subsystem clock frequency

(6) 20 MHz internal high-speed oscillation control register (DSCCTL)

This register controls the 20 MHz internal high-speed oscillation clock (DSC) function.

This register can be used to control oscillation of the 20 MHz internal high-speed oscillation clock (f_{IH20}) and select the 20 MHz internal high-speed oscillation clock (f_{IH20}) as the CPU/peripheral hardware clock.

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

Reset signal generation clears this register to 00H.

Figure 5-7. Format of 20 MHz Internal High-Speed Oscillation Control Register (DSCCTL)

Address: F00F6H After reset: 00H R/W[Note]

Symbol	7	6	5	4	<3>	<2>	1	<0>
DSCCTL	0	0	0	0	DSCS	SELDSC	0	DSCON

DSCS	20 MHz internal high-speed oscillation supply status flag
0	Not supplied
1	Supplied (The CPU/peripheral hardware clock (f_{CLK}) operates on the 20 MHz internal high-speed oscillation clock.)

SELDSC	Selection of 20 MHz internal high-speed oscillation for CPU/peripheral hardware clock (f_{CLK})
0	Does not select 20 MHz internal high-speed oscillation (clock selected by CKC register is supplied to f_{CLK})
1	Selects 20 MHz internal high-speed oscillation (20 MHz internal high-speed oscillation is supplied to f_{CLK})

DSCON	Operating or stopping 20 MHz internal high-speed oscillation clock (f_{IH20})
0	Stopped
1	Operated

Note Bit 3 is read-only.

Cautions 1. Set SELDSC when 100 μs have elapsed after having set DSCON.

2. The internal high-speed oscillator must be operated (HIOSTOP = 0) when DSCON = 1.

(7) 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.

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

- Real-time counter
- A/D converter
- Serial interface IICA
- Serial array unit SAU0 and SAU1
- Comparators/programmable gain amplifiers
- Timer array unit TAU0 and TAU1
- Remote controller receiver
- CEC transmission/reception circuit

Figure 5-8. Format of Peripheral Enable Registers (1/3)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

Address: F00F1H After reset: 00H R/W

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

RTCEN	Control of real-time counter (RTC) input clock supply [Note]
0	Stops input clock supply. • SFR used by the real-time counter (RTC) cannot be written. • Operation of the real-time counter (RTC) is in the reset status.
1	Supplies input clock. • SFR used by the real-time counter (RTC) can be read and written.

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

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

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 bit 6 of the PER0 register, and bits 0 to 3, 6, and 7 of the PER1 register to 0.

Figure 5-8. Format of Peripheral Enable Registers (2/3)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

Address: F00F1H After reset: 00H R/W

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

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

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

TAU1EN	Control of timer array unit 1 input clock supply
0	Stops input clock supply. • SFR used by timer array unit 1 cannot be written. • Timer array unit 1 is in the reset status.
1	Supplies input clock. • SFR used by timer array unit 1 can be read and written.

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

Caution Be sure to clear bit 6 of the PER0 register, and bits 0 to 3, 6, and 7 of the PER1 register to 0.

Figure 5-8. Format of Peripheral Enable Registers (3/3)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

Address: F00F1H After reset: 00H R/W

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

REMEN	Control of remote controller receiver input clock supply
0	Stops input clock supply. • SFR used by remote controller receiver cannot be written. • Remote controller receiver is in the reset status.
1	Supplies input clock. • SFR used by remote controller receiver can be read and written.

CECEN	Control of CEC transmission/reception circuit input clock supply
0	Stops input clock supply. • SFR used by CEC transmission/reception circuit cannot be written. • CEC transmission/reception circuit is in the reset status.
1	Supplies input clock. • SFR used by CEC transmission/reception circuit can be read and written.

Caution Be sure to clear bit 6 of the PER0 register, and bits 0 to 3, 6, and 7 of the PER1 register to 0.

(8) Operation speed mode control register (OSMC)

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

The FLPC and FSEL bits can be used to control the step-up circuit of the flash memory for high-speed operation.

If the microcontroller operates on a system clock of 10 MHz or more, set this register to 01B.

If the microcontroller operates at low speed on a system clock of 10 MHz or less, power consumption can be reduced, because the voltage booster can be stopped by setting this register to its initial value, 00B. Furthermore, when CPU operates with the system clock of 1 MHz, the power consumption can be further reduced by setting the FLPC bit to 1.

If the RTCLPC bit is set to 1, current consumption can be reduced, because the circuit that synchronizes the clock to the peripheral functions, except the real-time counter, is stopped in HALT mode while subsystem clock is selected as CPU clock.

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

Reset signal generation clears this register to 00H.

Figure 5-9. 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	RTCLPC	0	0	0	0	0	FLPC	FSEL

FLPC	FSEL	f_{CLK} frequency selection
0	0	Operates at a frequency of 10 MHz or less (default).
0	1	Operates at a frequency higher than 10 MHz.
1	0	Operates at a frequency of 1 MHz.
1	1	Setting prohibited

RTCLPC	Setting in subsystem clock HALT mode
0	Enables supply of subsystem clock to peripheral functions (See Table 19-1 for peripheral functions whose operations are enabled.)
1	Stops supply of subsystem clock to peripheral functions other than real-time counter

Cautions 1. Write "1" to the FSEL bit before the following two operations.
- **Changing the clock prior to dividing f_{CLK} to a clock other than f_{IH}.**
- **Operating the DMA controller.**

2. The CPU waits (140.5 clock (f_{CLK})) when "1" is written to the FSEL bit.
Interrupt requests issued during a wait will be suspended.
However, counting the oscillation stabilization time of f_X can continue even while the CPU is waiting.

3. To increase f_{CLK} to 10 MHz or higher, set the FSEL bit to "1", then change f_{CLK} after three or more clocks have elapsed.

4. To set the FSEL bit to 0, set f_{CLK} to 10 MHz or less in advance.

5. The HALT mode current when the subsystem clock is used can be reduced by setting the RTCLPC bit to 1. However, no clock can be supplied to the peripheral functions other than the real-time counter during HALT mode while subsystem clock is selected as CPU clock. Set bit 7 (RTCEN) of peripheral enable registers 0 (PER0) to 1, and all of bits 0 to 6 of the PER0 register and bits 0 to 7 of peripheral enable register 1 (PER1) to 0 before setting subsystem clock HALT mode.

6. If the FLPC bit is set to a frequency of 1 MHz or less and then set (1), it cannot be cleared (0) or set to a frequency of more than 1 MHz.

7. Flash memory can be used at a frequency of 10 MHz or lower if FSEL is 1.

5.4 System Clock Oscillator

5.4.1 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

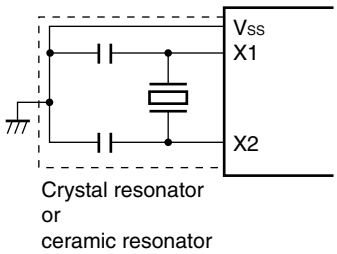
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

When the pins are not used as input port pins, either, see **Table 2-3 Connection of Unused Pins (78K0R/KF3-C)** or **Table 2-4 Connection of Unused Pins (78K0R/KG3-C)**.

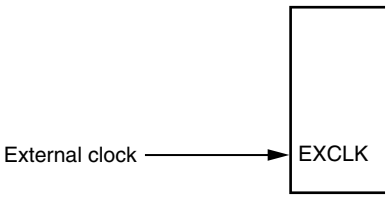
Figure 5-10 shows an example of the external circuit of the X1 oscillator.

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

(a) Crystal or ceramic oscillation



(b) External clock



Cautions are listed on the next page.

5.4.2 XT1 oscillator

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

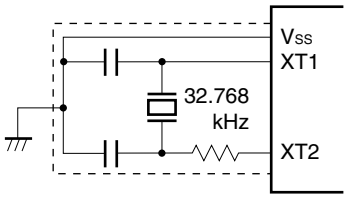
To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

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 2-3 Connection of Unused Pins (78K0R/KF3-C)** or **Table 2-4 Connection of Unused Pins (78K0R/KG3-C)**.

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

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

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

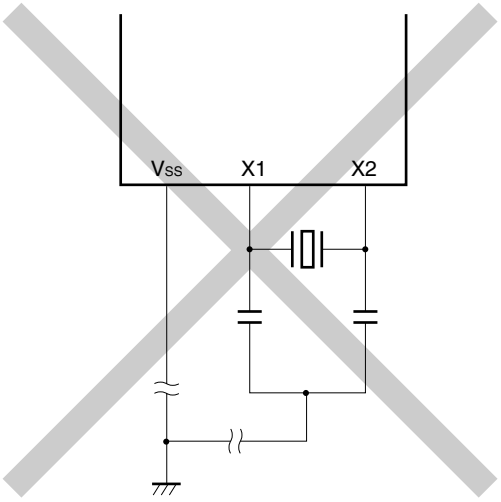
The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **When using the ultra-low power consumption oscillation (AMPHS1 = 1) as the mode of the XT1 oscillator, use the recommended resonators described in CHAPTER 29 ELECTRICAL SPECIFICATIONS.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1 = 1) is selected.**
- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential as V_{SS} as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

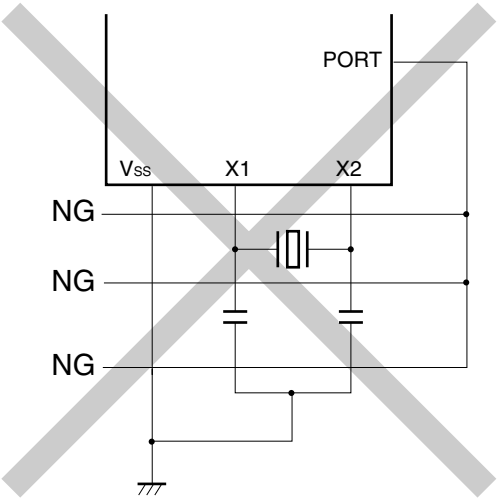
Figure 5-12 shows examples of incorrect resonator connection.

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

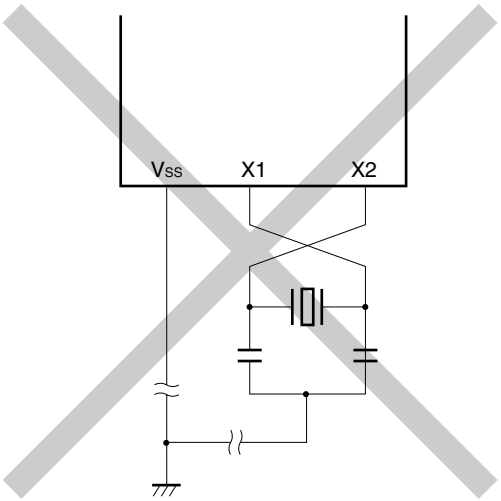
(a) Too long wiring



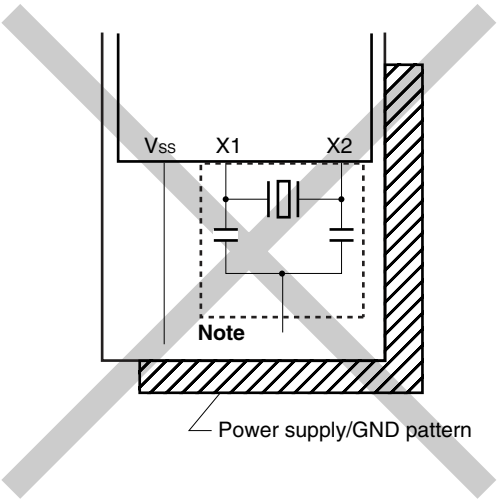
(b) Crossed signal line



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



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

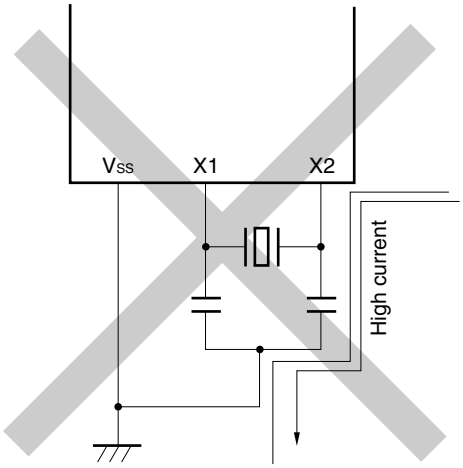


Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

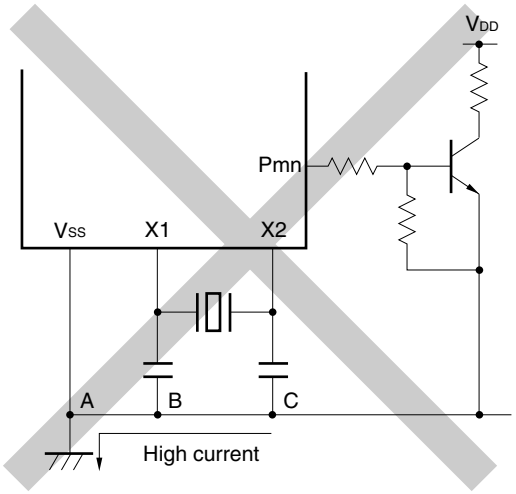
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

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

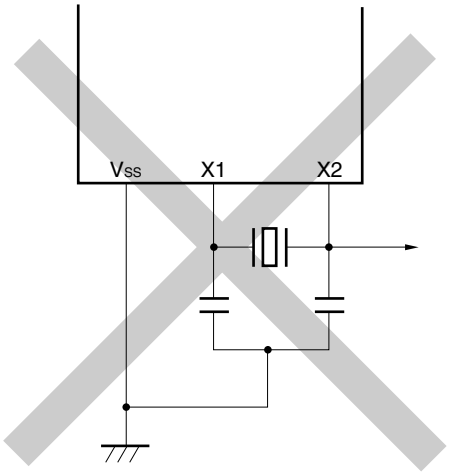
(e) Wiring near high alternating current



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



(g) Signals are fetched



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

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

5.4.3 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0R/Kx3-C (8 and 20 MHz (TYP.)). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC) and bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL).

After a reset release, the 8 MHz internal high-speed oscillator automatically starts oscillating after reset release. 20 MHz internal high-speed oscillator is started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1.

5.4.4 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0R/Kx3-C.

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 (30 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.

5.4.5 Prescaler

The prescaler generates a CPU/peripheral hardware clock by dividing the main system clock and subsystem clock.

5.5 Clock Generator Operation

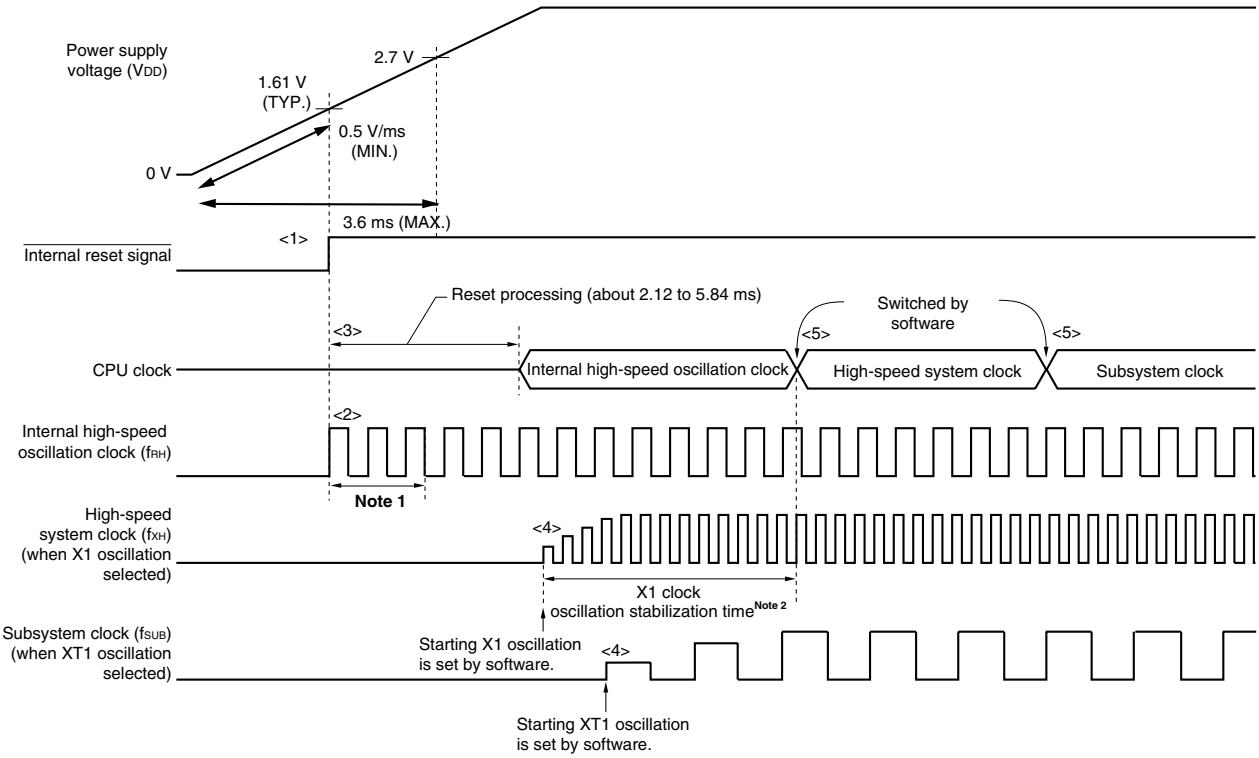
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 5-1**).

- Main system clock f_{MAIN}
  - High-speed system clock f_{MX}
    X1 clock f_X
    External main system clock f_{EX}
  - Internal high-speed oscillation clock f_{IH}
  - 20 MHz internal high-speed oscillation clock f_{IH20}
- 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/Kx3-C.

When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-13 to Figure 5-16.

Figure 5-13. 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 1.61 V (TYP.), the reset is released and the internal high-speed oscillator automatically starts oscillation.

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

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **(1)** in **5.6.1 Example of controlling high-speed system clock** and **(1)** in **5.6.3 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 **5.6.1 Example of controlling high-speed system clock** and **(2)** in **5.6.3 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).

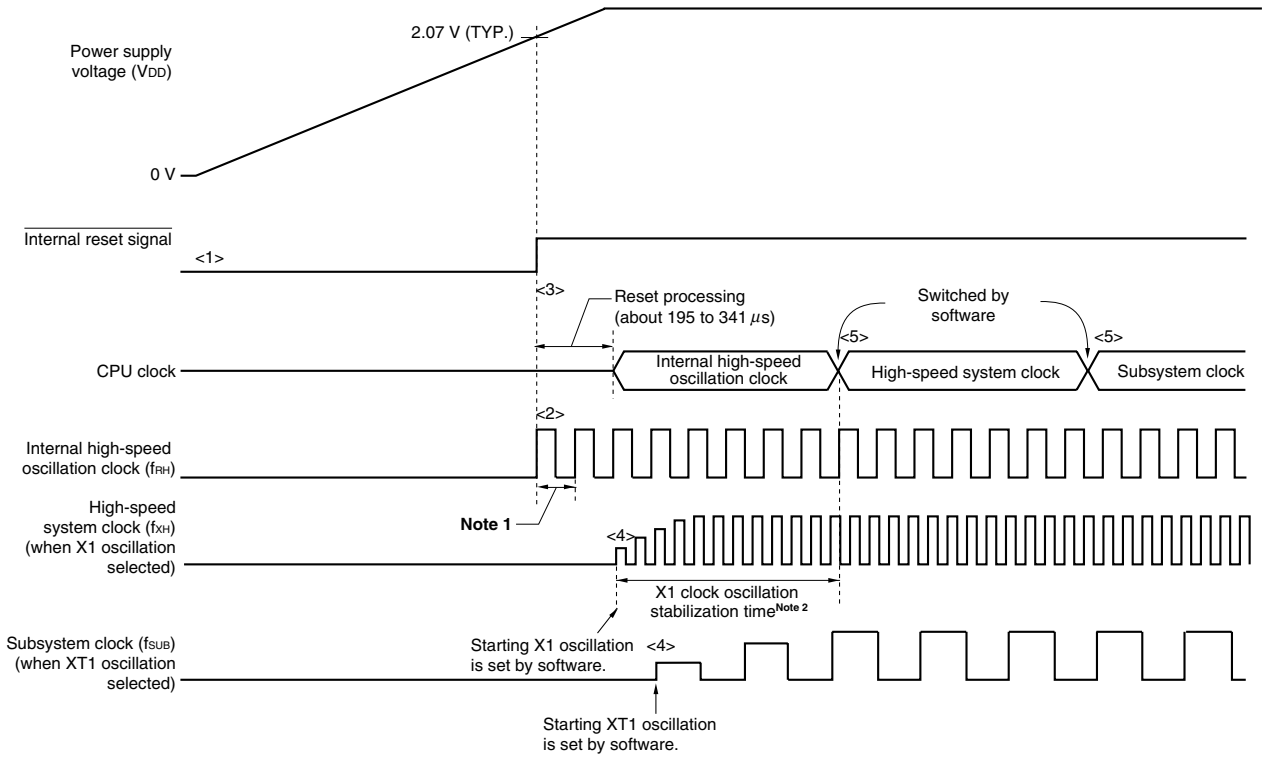
Cautions 1. Set so that no more than 3.6 ms elapses between when power is applied and when the voltage reaches 2.7 V. If more time is required (if the voltage needs to rise more slowly than the 0.5 V/ms (MIN.) rating), be sure to input a low level to the $\overline{\text{RESET}}$ pin before the voltage reaches 2.7 V after power application or set the LVI default start function stopped by using the option byte (LVIOFF = 0) (see Figure 5-14). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 5-13 after reset release by the $\overline{\text{RESET}}$ pin.

Cautions 2. It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.

3. Some operations can also be executed while V_{DD} < 2.7 V (For details, figures of CHAPTER 29 ELECTRICAL SPECIFICATIONS AC Characteristics (1) Basic operation.

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 **5.6.1 Example of controlling high-speed system clock**, **(3)** in **5.6.2 Example of controlling internal high-speed oscillation clock**, and **(3)** in **5.6.3 Example of controlling subsystem clock**).

Figure 5-14. 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) circuit.

<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 **5.6.1 Example of controlling high-speed system clock** and **(1)** in **5.6.3 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 **5.6.1 Example of controlling high-speed system clock** and **(2)** in **5.6.3 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).

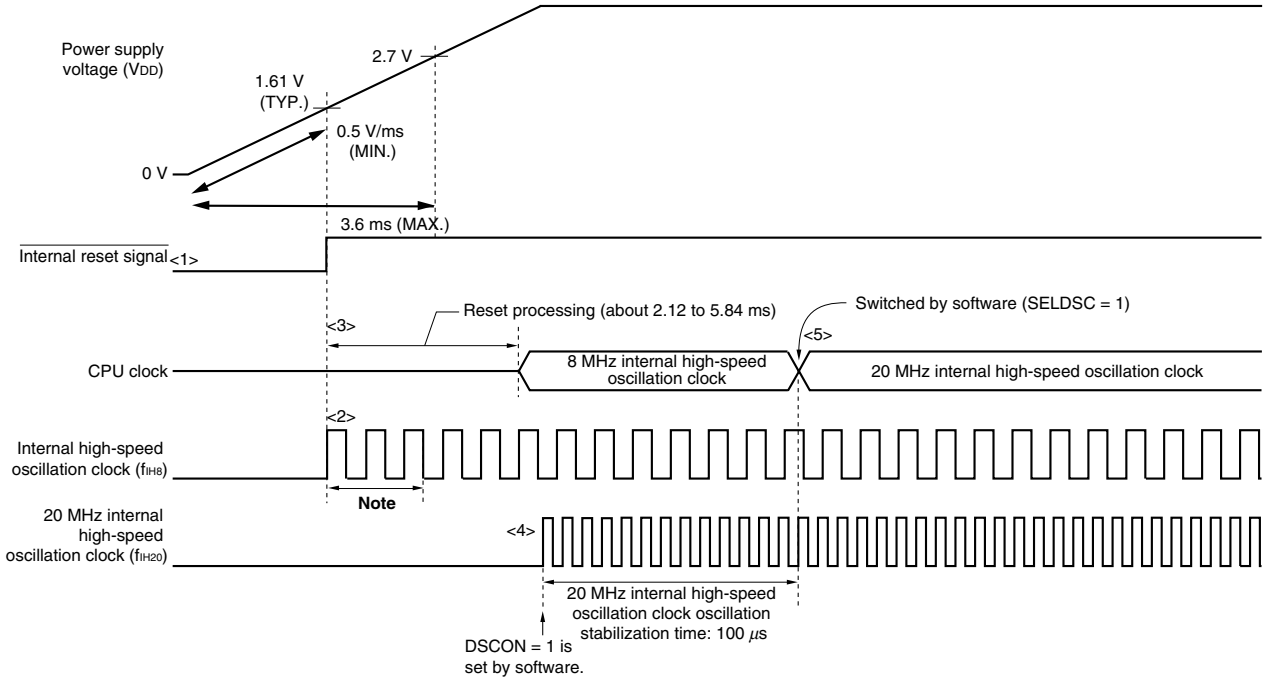
Cautions 1. A voltage stabilization time (about 2.12 to 5.84 ms) is required after the supply voltage reaches 1.61 V (TYP.). If the time for the supply voltage to rise from 1.61 V (TYP.) to 2.07 V (TYP.) is shorter than the voltage stabilization time, reset processing is entered after the voltage stabilization time elapses.

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

Caution 3. Some operations can also be executed while V_{DD} < 2.7 V (For details, figures of CHAPTER 29 ELECTRICAL SPECIFICATIONS AC Characteristics (1) Basic operation.

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 **5.6.1 Example of controlling high-speed system clock, (3)** in **5.6.2 Example of controlling internal high-speed oscillation clock**, and **(3)** in **5.6.3 Example of controlling subsystem clock**).

Figure 5-15. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1) and Changing to 20 MHz Internal High-Speed Oscillation Clock)



<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 1.61 V (TYP.), the reset is released and the internal high-speed oscillator automatically starts oscillation.

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

<4> Set DSCON = 1 by software.

<5> Switch the clock by setting SELDSC = 1 by software after waiting for 100 μs.

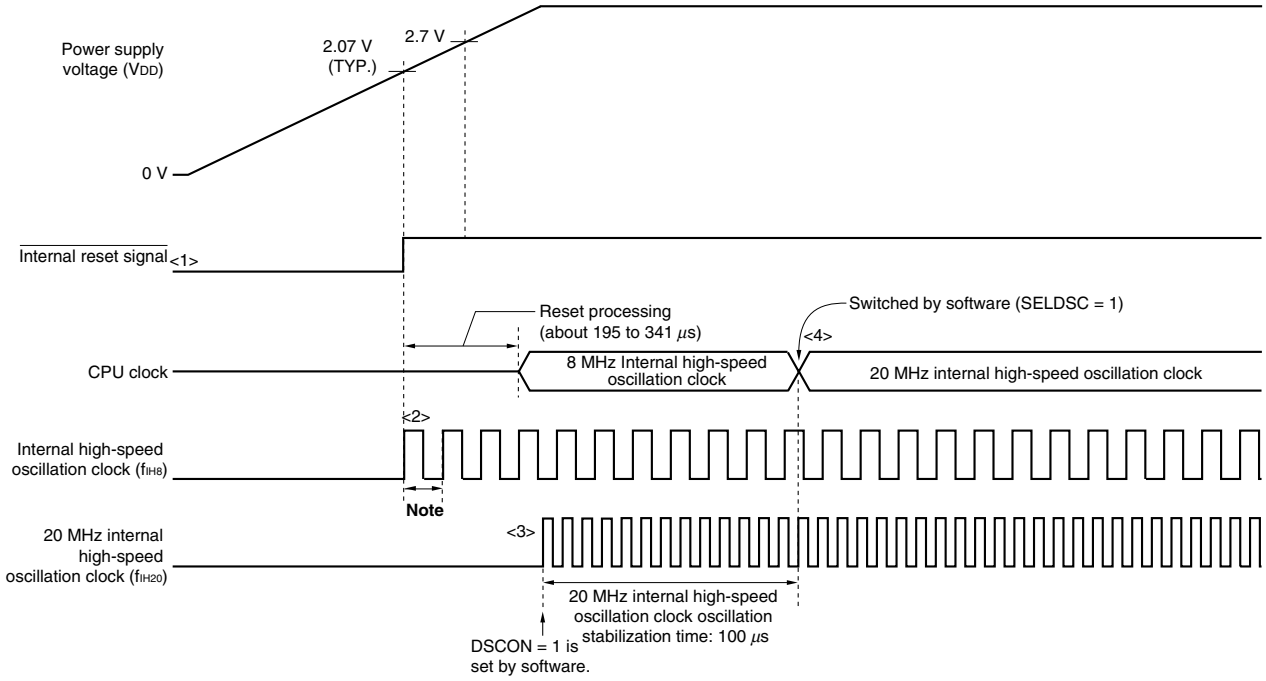
Note The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

Cautions 1. To use the 20 MHz internal high-speed oscillation clock, use bits 2 and 1 (FRQSEL2 and FRQSEL1) of the option byte (000C1H) to set the frequency to 20 MHz in advance (for details, see CHAPTER 24 OPTION BYTE).

Cautions 2. Set so that no more than 3.6 ms elapses between when power is applied and when the voltage reaches 2.7 V. If more time is required (if the voltage needs to rise more slowly than the 0.5 V/ms (MIN.) rating), be sure to input a low level to the $\overline{\text{RESET}}$ pin before the voltage reaches 2.7 V after power application or set the LVI default start function stopped by using the option byte (LVIOFF = 0) (see Figure 5-16). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 5-15 after reset release by the $\overline{\text{RESET}}$ pin.

3. Some operations can also be executed while V_{DD} < 2.7 V (For details, figures of CHAPTER 29 ELECTRICAL SPECIFICATIONS AC Characteristics (1) Basic operation.

Figure 5-16. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0) and Changing to 20 MHz Internal High-Speed Oscillation Clock)



<1> When the power is turned on, an internal reset signal is generated by the low-voltage detector (LVI) circuit.

<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 DSCON = 1 by software.

<5> Switch the clock by setting SELDSC = 1 by software after waiting for 100 μs.

Note The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

Cautions 1. To use the 20 MHz internal high-speed oscillation clock, use bits 2 and 1 (FRQSEL2 and FRQSEL1) of the option byte (000C1H) to set the frequency to 20 MHz in advance (for details, see CHAPTER 24 OPTION BYTE).

Cautions 2. A voltage stabilization time (about 2.12 to 5.84 ms) is required after the supply voltage reaches 1.61 V (TYP.). If the time for the supply voltage to rise from 1.61 V (TYP.) to 2.07 V (TYP.) is shorter than the voltage stabilization time, reset processing is entered after the voltage stabilization time elapses.

3. Some operations can also be executed while V_{DD} < 2.7 V (For details, figures of CHAPTER 29 ELECTRICAL SPECIFICATIONS AC Characteristics (1) Basic operation.

5.6 Controlling Clock

5.6.1 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 MHz ≤ fx ≤ 10 MHz

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0	1	0	0/1	0	0/1	0/1	0

- 10 MHz < fx ≤ 20 MHz

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0	1	0	0/1	0	0/1	0/1	1

Remarks 1. fx: X1 clock oscillation frequency

2. For setting of the P123/XT1 and P124/XT2 pins, see **5.6.3 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 5.6.3 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).

(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	AMPHS1	AMPHS0	AMPH
1	1	0	0/1	0	0/1	0/1	0/1

Remark For setting of the P123/XT1 and P124/XT2 pins, see **5.6.3 (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 5.6.3 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).

(3) Example of setting procedure when using high-speed system clock as CPU/peripheral hardware clock

<1> Setting high-speed system clock oscillation[Note]
(See **5.6.1 (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_{CLK})
1	0	0	0	f_{MX}
	0	0	1	$f_{MX}/2$
	0	1	0	$f_{MX}/2^2$
	0	1	1	$f_{MX}/2^3$
	1	0	0	$f_{MX}/2^4$
	1	0	1	$f_{MX}/2^5$ [Note]

Note Setting is prohibited when f_{MX} < 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	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

(PER1 register)

0	0	REMEN	CECEN	0	0	0	0

xxxEN	Input clock control
0	Stops input clock supply.
1	Supplies input clock.

Caution Be sure to clear bit 6 of PER0 register and bits 0 to 3, 6, and 7 of PER1 to 0.

Remark RTCEN: Control of the real-time counter input clock
ADCEN: Control of the A/D converter input clock
IICAEN: Control of the serial interface IICA 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
REMEN: Control of the remote controller receiver input clock
CECEN: Control of the CEC transmission/reception circuit 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 or 20 MHz 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[Note]

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.

5.6.2 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[Note]

<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[Note]

(See **5.6.2 (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_{CLK})
0	0	0	0	f_{IH}
	0	0	1	$f_{IH}/2$
	0	1	0	$f_{IH}/2^2$
	0	1	1	$f_{IH}/2^3$
	1	0	0	$f_{IH}/2^4$
	1	0	1	$f_{IH}/2^5$

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 or 20 MHz 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.

5.6.3 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 IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS.

(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	AMPHS1	AMPHS0	AMPH
0/1	0/1	0	1	0	0/1	0/1	0/1

Remark For setting of the P121/X1 and P122/X2 pins, see **5.6.1 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 5.6.1 (1) Example of setting procedure when oscillating the X1 clock or 5.6.1 (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[Note]
(See **5.6.3 (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 IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS).

(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.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock or 20 MHz 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.

5.6.4 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 (30 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.

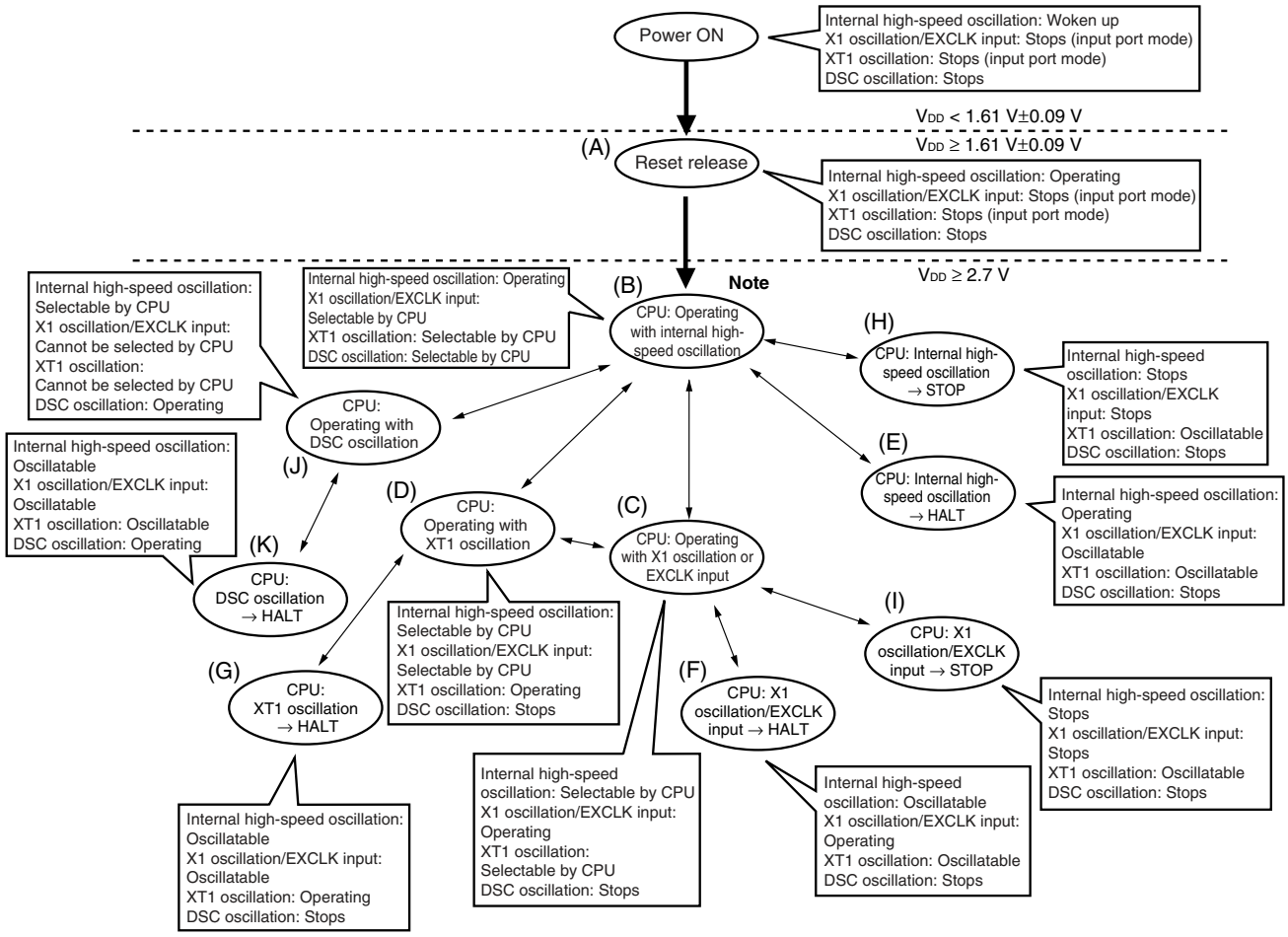
- Release the HALT or STOP mode

  (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).

5.6.5 CPU clock status transition diagram

Figure 5-17 shows the CPU clock status transition diagram of this product.

Figure 5-17. CPU Clock Status Transition Diagram



Note After reset release, operation at 4 MHz (8 MHz/2) is started, because $f_{CLK} = f_{IH}/2$ has been selected by setting the system clock control register (CKC) to 09H.

Remarks 1. 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±0.2 V[Note].

After the reset operation, the status will shift to (B) in the above figure.

2. DSC: 20 MHz internal high-speed oscillation clock

Table 5-4 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (1/6)

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

Notes 1. The CMC register can be written only once by an 8-bit memory manipulation instruction after reset release.

2. FSEL = 1 when f_{CLK} > 10 MHz
If a divided clock is selected and f_{CLK} ≤ 10 MHz, use with FSEL = 0 is possible even if fx > 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).

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 Note			CSC Register	Waiting for Oscillation Stabilization	CKC Register
	OSCSELS	AMPHS1	AMPHS0	XTSTOP		CSS
(A) → (B) → (C)	1	0/1	0/1	0	Must be checked	1

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

Remark (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-17.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (2/6)

(4) CPU operating with 20 MHz internal high-speed oscillation clock (J) 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	DSCCTL Register	Waiting for Oscillation Stabilization	DSCCTL Register
	DSCON		SELDSC
(A) → (B) → (J)	1	Necessary (100 μs)	1

(5) 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[Note 1]			OSTS Register	CSC Register	OSMC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP	FSEL		MCM0
(B) → (C) (X1 clock: 2 MHz ≤ fx ≤ 10 MHz)	0	1	0	**Note 2**	0	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	**Note 2**	0	1[Note 3]	Must be checked	1
(B) → (C) (external main clock)	1	1	×	**Note 2**	0	0/1[Note 3]	Must not be checked	1

Unnecessary if these registers are already set

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

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

2. Set the oscillation stabilization time as follows.

- Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS

3. FSEL = 1 when f_{CLK} > 10 MHz

If a divided clock is selected and f_{CLK} ≤ 10 MHz, use with FSEL = 0 is possible even if f_X > 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).

Remarks 1 ×: don't care

2 (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-17.

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

(6) 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[Note]	CSC Register	Waiting for Oscillation Stabilization	CKC Register
	OSCSELS	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.

(7) CPU clock changing from internal high-speed oscillation clock (B) to 20 MHz internal high-speed oscillation clock (J)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	DSCCTL Register	Waiting for Oscillation Stabilization	DSCCTL Register
	DSCON		SELDSC
(B) → (J)	1	Necessary (100 μs)	1

Unnecessary if the CPU is operating with the 20 MHz internal high-speed oscillation clock

(8) 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

Remark (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-17.

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

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

(Setting sequence of SFR registers) →

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

Unnecessary if the CPU is operating with the subsystem clock

(10) 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 (K) in Table 5-4 correspond to (A) to (K) in Figure 5-17.

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

(11) 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 MHz ≤ f_X ≤ 10 MHz)	**Note 1**	0	0	Must be checked	1	0
(D) → (C) (X1 clock: 10 MHz < f_X ≤ 20 MHz)	**Note 1**	0	1 Note 2	Must be checked	1	0
(D) → (C) (external main clock)	**Note 1**	0	0/1 Note 2	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 MHz
If a divided clock is selected and f_{CLK} ≤ 10 MHz, use with FSEL = 0 is possible even if f_X > 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).

(12) CPU clock changing from 20 MHz internal high-speed oscillation clock (J) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	DSCCTL Register	
	SELDSC	DSCON
(J) → (B)	0	0

Remark (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-17.

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

(13) • 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)
• HALT mode (K) set while CPU is operating with 20 MHz internal high-speed oscillation clock (J)

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

(14) • 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	
	In external clock		–	

Remark (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-17.

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

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

Table 5-5. 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 XT1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	
	20 MHz internal high-speed oscillation clock	Stabilization of DSC oscillation with 20 MHz set by using the option byte • After elapse of oscillation stabilization time (100 μs) after setting to DSCON = 1 • SELDSC = 1	–
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).
	20 MHz internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
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).
	20 MHz internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–

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

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Subsystem clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator and selection of internal high-speed oscillation clock as main system clock • HIOSTOP = 0, MCS = 0	XT1 oscillation can be stopped (XTSTOP = 1)
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	20 MHz internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
20 MHz internal high-speed oscillation clock	Internal high-speed oscillation clock	• SELDSC = 0 (Set when changing the clock.)	20 MHz internal high-speed oscillation clock can be stopped (DSCON = 0)
	X1 clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
	External main system clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
	Subsystem clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–

5.6.7 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 5-6 to Table 5-9).

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.

Table 5-6. Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{MAINC}	←→ (changing the division ratio)	f_{MAINC}	See Table 5-7
f_{IH}	←→	f_{MX}	See Table 5-8
f_{MAINC}	←→	$f_{SUB}/2$	See Table 5-9

Table 5-7. Maximum Number of Clocks Required for f_{MAINC} ↔ f_{MAINC} (Changing the Division Ratio)

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	

Table 5-8. Maximum Number of Clocks Required for f_{IH} ↔ f_{MX}

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

Table 5-9. Maximum Number of Clocks Required for $f_{MAINC} \leftrightarrow f_{SUB}$

Set Value Before Switchover	Set Value After Switchover	
CSS	CSS	
	0 ($f_{CLK} = f_{MAINC}$)	1 ($f_{CLK} = f_{SUB}/2$)
0 ($f_{CLK} = f_{MAINC}$)		1 + $4f_{MAINC}/f_{SUB}$ clock
1 ($f_{CLK} = f_{SUB}/2$)	2 + $f_{SUB}/2f_{MAINC}$ clock	

Remarks 1. f_{IH} :Internal high-speed oscillation clock frequency
f_{MX} :High-speed system clock frequency
f_{MAIN} :Main system clock frequency
f_{MAINC} :Main system select clock frequency
f_{SUB} :Subsystem clock frequency
f_{CLK} :CPU/peripheral hardware clock frequency

2. The number of clocks listed in Table 5-7 to Table 5-9 is the number of CPU clocks before switchover.

3. Calculate the number of clocks in Table 5-7 to Table 5-9 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_{IH} = 8 MHz, f_{MX} = 10 MHz)

$1 + f_{IH}/f_{MX} = 1 + 8/10 = 1 + 0.8 = 1.8 \rightarrow 2$ clocks

5.6.8 Conditions before clock oscillation is stopped

The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped.

Table 5-10. 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
20 MHz internal high-speed oscillation clock	SELDSC = 0 (The main system clock is operating on a clock other than the 20 MHz internal high-speed oscillation clock.)	DSCON = 0

CHAPTER 6 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 three 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
• Interval timer • Square wave output • External event counter • Divider function (channel 0 of unit 0 only) • Input pulse interval measurement • Measurement of high-/low-level width of input signal	• PWM output • One-shot pulse output • Multiple PWM output

6.1 Functions of Timer Array Unit

The timer array unit has the following functions.

6.1.1 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 **6.6.1 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.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 12

(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 12

6.1.2 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 **6.6.1 Overview of single-operation function and combination-operation function**).

(1) PWM (Pulse Width Modulation) output

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

(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 Modulation) output

By extending the PWM function and using one master channel and two or more slave channels, up to seven types of PWM signals that have a specific period and a specified duty factor can be generated.

6.2 Configuration of Timer Array Unit

The timer array unit includes the following hardware.

Table 6-1. 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 TI12 pins
Timer output	TO00 to TO07, TO10 to TO12 pins, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register m (TPSm) • Timer channel enable status register m (TEm) • Timer channel start register m (TSm) • Timer channel stop register m (TTm) • Timer input select register 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)
	<Registers of each channel> • Common Timer mode register mn (TMRmn) Timer status register mn (TSRmn) Noise filter enable registers 1, 2 (NFEN1, NFEN2) • 78K0R/KF3-C Port mode registers 0, 1, 3 to 6 (PM0, PM1, PM3 to PM6) Port registers 0, 1, 3 to 6 (P0, P1, P3 to P6) • 78K0R/KG3-C Port mode registers 0, 1, 3, 4, 6, 13, 14 (PM0, PM1, PM3, PM4, PM6, PM13, PM14) Port registers 0, 1, 3, 4, 6, 13, 14 (P0, P1, P3, P4, P6, P13, P14)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)
mn = 00 to 07, 10 to 12

Figure 6-1 and Figure6-2 show the block diagram.

Figure 6-1. Block Diagram of Timer Array Unit 0

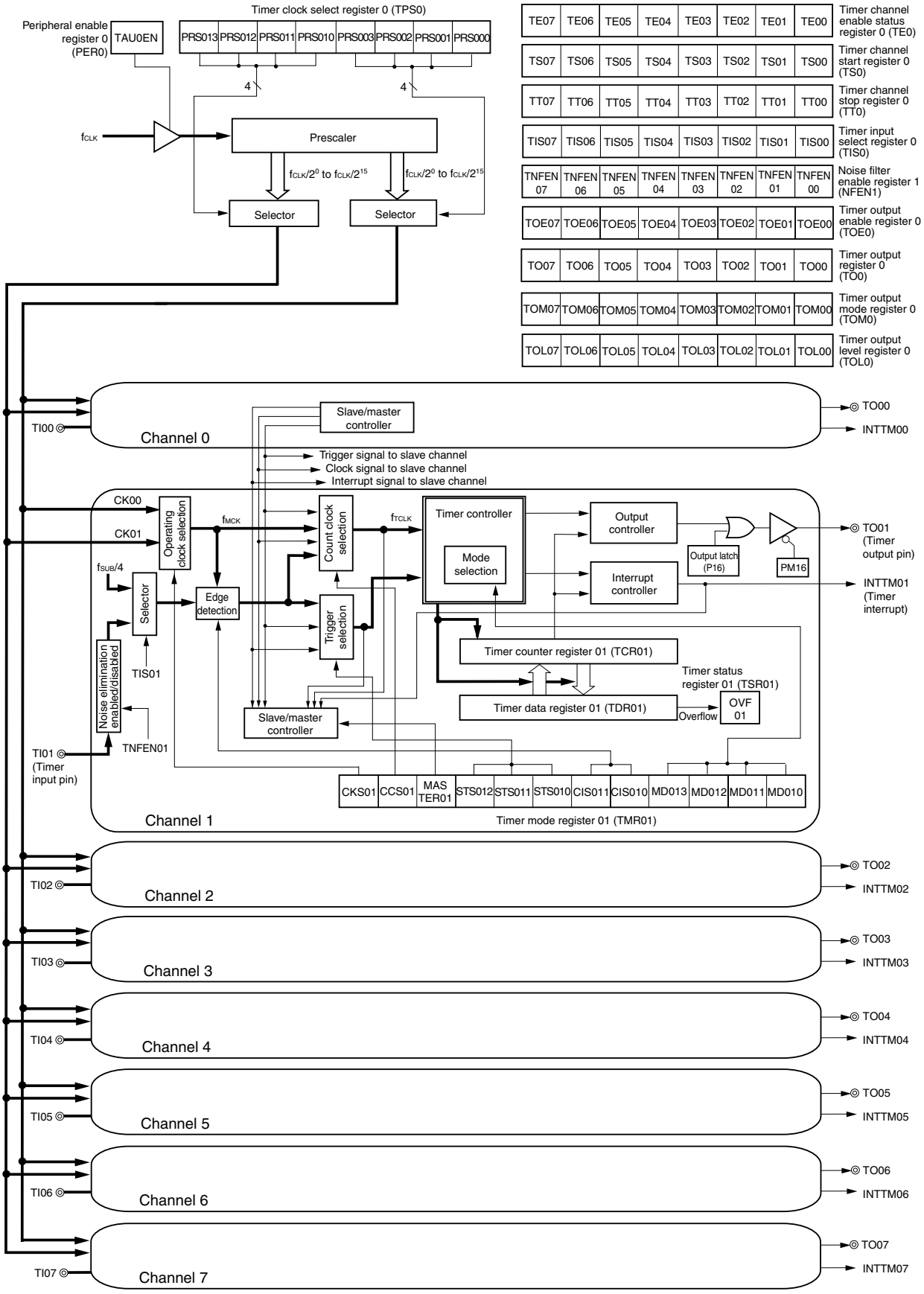
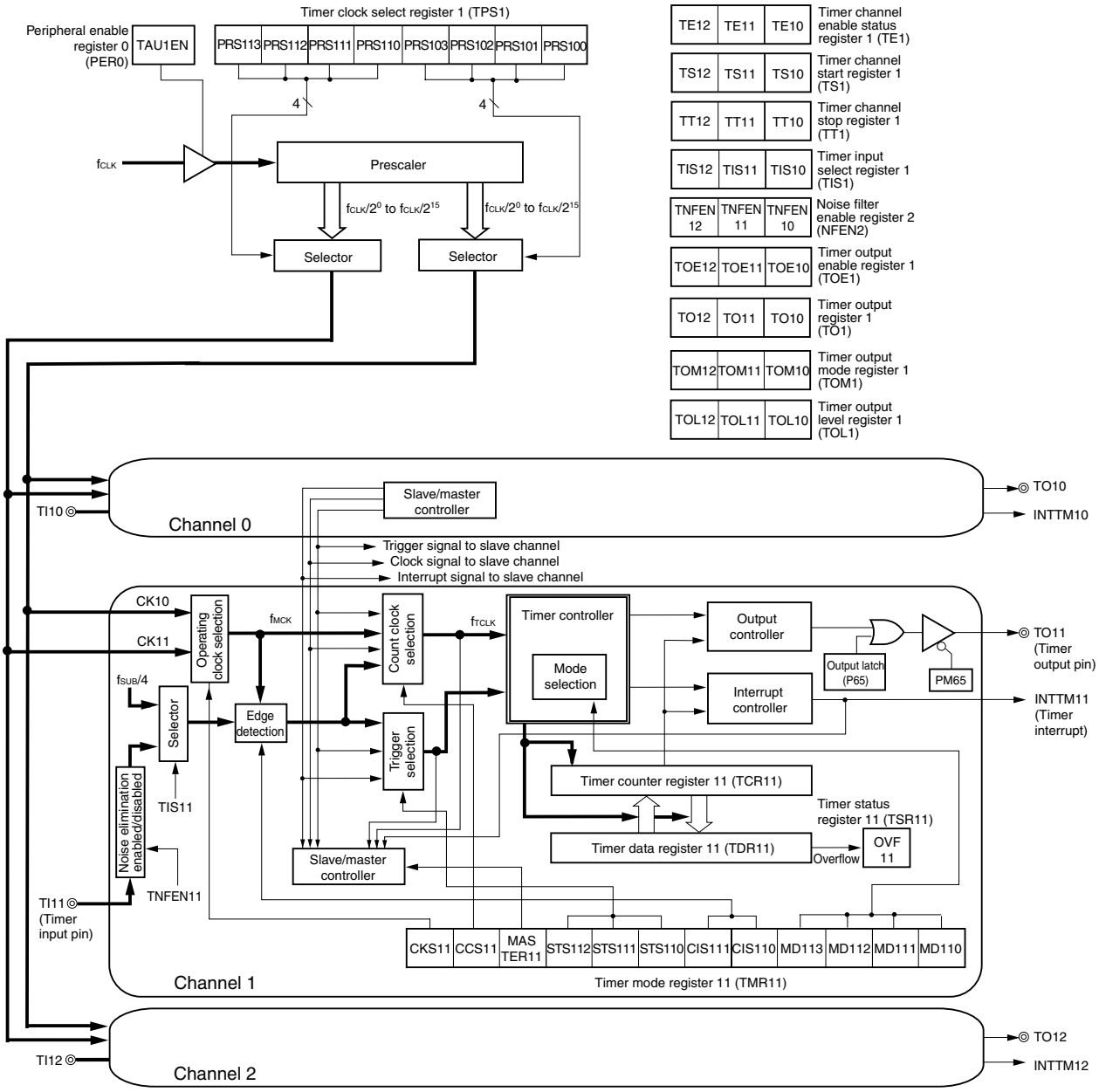


Figure 6-2. Block Diagram of Timer Array Unit 1



(1) Timer counter register mn (TCRmn)

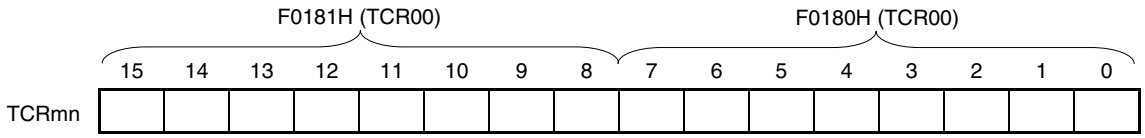
TCRmn is a 16-bit read-only register and is used to count clocks.

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 6-3. Format of Timer Counter Register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07), F01C0H, F01C1H (TCR10) to F01C4H, F01C5H (TCR12) After reset: FFFFH R



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 12

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

Table 6-2. TCRmn Register Read Value in Various Operation Modes

Operation Mode	Count Mode	TCRmn Register Read Value[Note]			
		Operation mode change after reset	Operation mode change after count operation paused (TTmn = 1)	Operation restart after count operation paused (TTmn = 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 TDRmn register + 1

Note The read values of the TCRmn register when TSmn has been set to "1" while TEmn = 0 are shown. The read value is held in the TCRmn register until the count operation starts.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 12

(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.

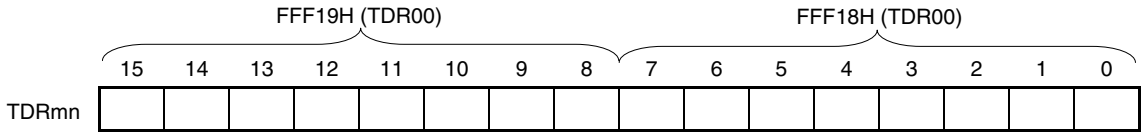
The value of TDRmn can be changed at any time.

This register can be read or written in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 6-4. Format of Timer Data Register mn (TDRmn)

Address: FFF18H, FFF19H (TDR00), FFF1AH, FFF1BH (TDR01), FFF64H, FFF65H (TDR02) to FFF6EH, FFF6FH (TDR07), FFF70H, FFF71H (TDR10) to FFF74H, FFF75H (TDR12) After reset: 0000H R/W



(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 (INTTMmn) 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 12

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer input select register 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)
- Noise filter enable registers 1, 2 (NFEN1, NFEN2)
- Port mode registers (PMxx)
- Port registers (Pxx)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

(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 6-5. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

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

Cautions 1. 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 noise filter enable registers 1, 2 (NFEN1, NFEN2), port mode registers (PMxx), and port registers (Pxx)).

2. Be sure to clear bit 6 of the PER0 register to 0.

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.

Figure 6-6. 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) Note				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz
0	1	1	1	$f_{CLK}/2^7$	15.62 kHz	39.1 kHz	78.1 kHz	156.2 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.76 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	976 Hz	2.44 kHz	4.88 kHz	9.76 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop the timer array unit (TTm = 00FFH).
The timer array unit must also be stopped if the operating clock specified by using the CKSmn bit (f_{MCK}), the valid edge of the signal input from the TImn pin, or the subsystem clock divided by 4 ($f_{SUB}/4$) is selected as the count clock (f_{TCLK}).

Caution Be sure to clear bits 15 to 8 to "0".

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency
2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), k = 0, 1, mn = 00 to 07, 10 to 12

(3) Timer mode register mn (TMRmn)

TMRmn sets an operation mode of channel n. It is used to select an operation clock (f_{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 **6.7 Operation of Timer Array Unit as Independent Channel** and **6.8 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 6-7. 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 F01CCH, F01CDH (TMR12)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn	0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by TPSm register
1	Operation clock CKm1 set by TPSm register
Operation clock f_{MCK} is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCSmn bit.	

CCSmn	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock f_{MCK} specified by CKSmn bit
1	Valid edge of input signal input from TImn pin/subsystem clock divided by four (f_{SUB}/4)
Count clock f_{TCLK} is used for the timer/counter, output controller, and interrupt controller.	

Cautions 1. Be sure to clear bits 14, 13, 5, and 4 to "0".

2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for f_{CLK} is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKSmn bit (f_{MCK}), the valid edge of the signal input from the TImn pin, or the subsystem clock divided by 4 (f_{SUB}/4) is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 12

Figure 6-7. 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 F01CCH, F01CDH (TMR12)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn	0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

MASTERmn	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 odd-numbered channels as slave channels (MASTERmn = 0). Clear the MASTERmn bit to 0 for a channel that is used with the independent channel operation function.	

STSmn2	STSmn1	STSmn0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of 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

CISmn1	CISmn0	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 12

Figure 6-7. 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 F01CCH, F01CDH (TMR12)

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 bit 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
• Interval timer mode (0, 0, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• Capture mode (0, 1, 0)	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode Note 1 (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
	1	Start trigger is valid during counting operation Note 2. At that time, interrupt is also generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time interrupt is not generated, either.
Other than above		Setting prohibited

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

2. 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 12

(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 6-3 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 6-8. Format of Timer Status Register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07), After reset: 0000H R
F01D0H, F01D1H (TSR10) to F01D4H, F01D5H (TSR12)

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 12

Table 6-3. 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 • Event counter mode • One-count mode	clear	–
	set	(Use prohibited, not set and not cleared)

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 (TEm)

TEm 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.

TEm can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TEm can be set with a 1-bit or 8-bit memory manipulation instruction with TEmL.

Reset signal generation clears this register to 0000H.

Figure 6-9. Format of Timer Channel Enable Status Register m (TEm)

Address: F01B0H, F01B1H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	0	TE12	TE11	TE10

TE mn	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 12

(6) Timer channel start register m (TSm)

TSm is a trigger register that is used to clear a timer counter (TCRmn) and start the counting operation of each channel.

When a bit (TSmn) of this register is set to 1, the corresponding bit (TEmn) of timer channel enable status register m (TEm) is set to 1. TSmn is a trigger bit and cleared immediately when TEmn = 1.

TSm can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TSm can be set with a 1-bit or 8-bit memory manipulation instruction with TSmL.

Reset signal generation clears this register to 0000H.

Figure 6-10. Format of Timer Channel Start Register m (TSm)

Address: F01B2H, F01B3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	0	TS12	TS11	TS10

TS mn	Operation enable (start) trigger of channel n
0	No trigger operation
1	TEmn is set to 1 and the count operation becomes enabled. The TCRmn count operation start in the count operation enabled state varies depending on each operation mode (see Table 6-4).

Caution Be sure to clear bits 15 to 8 of TS0 and bits 15 to 3 of TS1 to "0"

Remarks 1. When the TSm 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 12

Table 6-4. Operations from Count Operation Enabled State to TCRmn Count Start (1/2)

Timer operation mode	Operation when TSmn = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TSmn=1) until count clock generation. The first count clock loads the value of TDRmn to TCRmn and the subsequent count clock performs count down operation (see **6.3 (6) (a) Start timing in interval timer mode**).
• Event counter mode	Writing 1 to TSmn bit loads the value of TDRmn to TCRmn. The subsequent count clock performs count down operation. The external trigger detection selected by STSmn2 to STSmn0 bits in the TMRmn register does not start count operation (see **6.3 (6) (b) Start timing in event counter mode**).
• Capture mode	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 **6.3 (6) (c) Start timing in capture mode**).

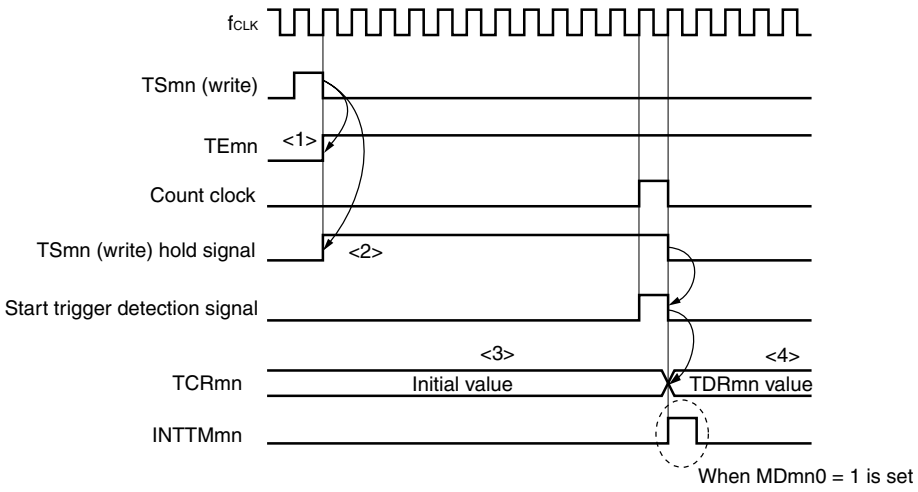
Table 6-4. Operations from Count Operation Enabled State to TCRmn Count Start (2/2)

Timer operation mode	Operation when TSmn = 1 is set
• One-count mode	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 **6.3 (6) (d) Start timing in one-count mode**).
• Capture & one-count mode	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 **6.3 (6) (e) Start timing in capture & one-count mode**).

(a) Start timing in interval timer 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, the “TDRmn value” is loaded to TCRmn and count starts.

Figure 6-11. Start Timing (In Interval Timer Mode)

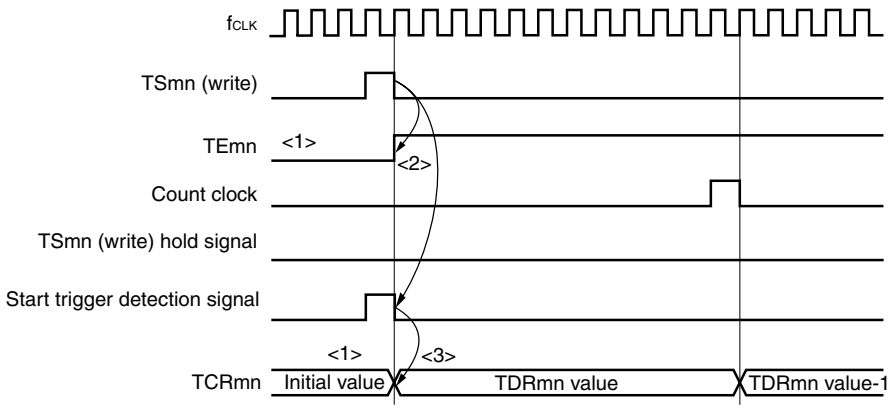


Caution In the first cycle operation of count clock after writing TSmn, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 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.

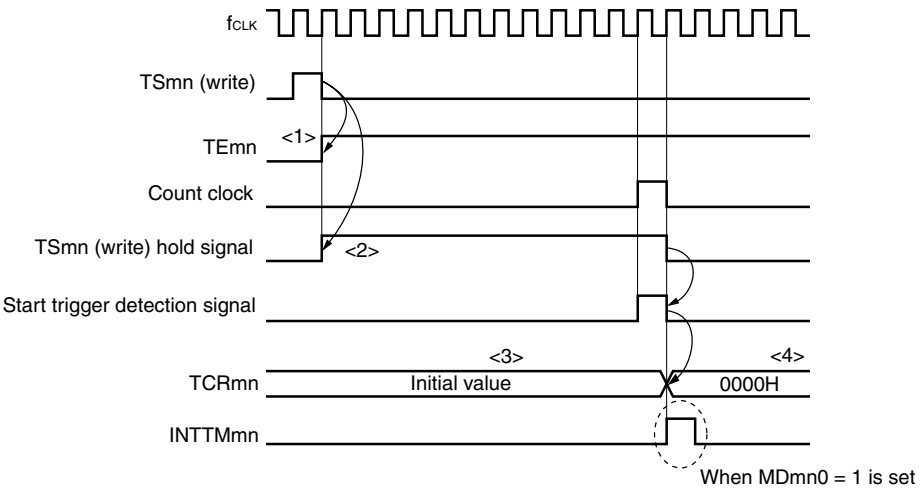
Figure 6-12. Start Timing (In Event Counter Mode)



(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 6-13. 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.

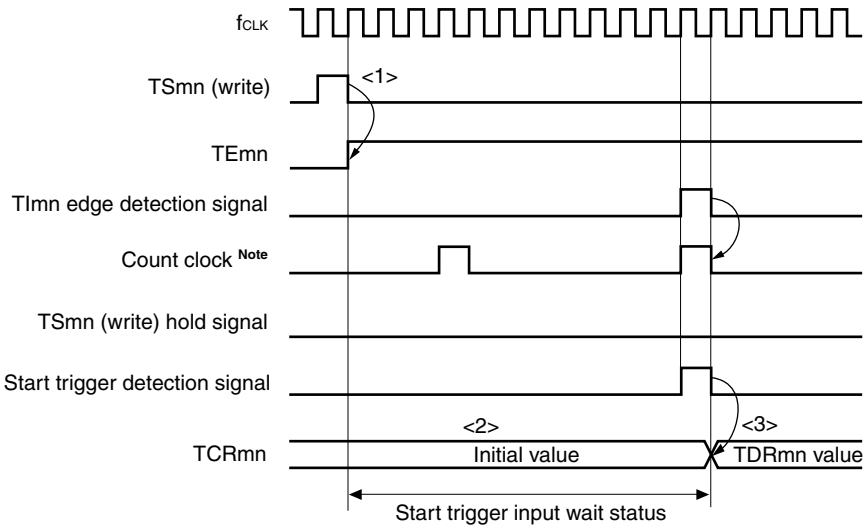
(d) Start timing in one-count mode

<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 6-14. Start Timing (In One-count Mode)



Note When the one-count mode is set, the operation clock (f_{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).

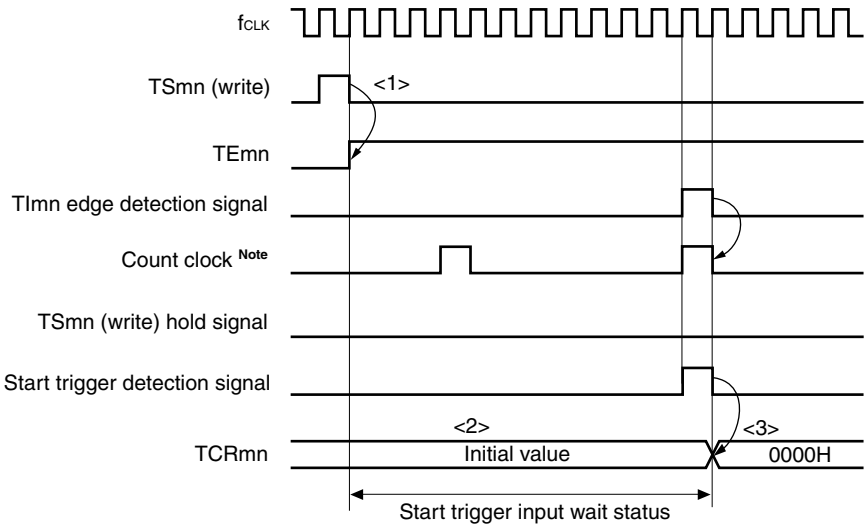
(e) Start timing in capture & one-count mode

<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 6-15. Start Timing (In Capture & One-count Mode)



Note When the capture & one-count mode is set, the operation clock (f_{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 6-16. 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	0	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 3 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 12

(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 6-17. 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	0	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 12

(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 6-18. Format of Timer Output Enable Register m (TOEm)

Address: F01BAH, F01BBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE0	0	0	0	0	0	0	0	0	TOE07	TOE06	TOE05	TOE04	TOE03	TOE02	TOE01	TOE00

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	0	TOE12	TOE11	TOE10

TOEmn	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 TOmn bit 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 3 of TOE1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

(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 following pin as a port function pin, set the corresponding TOmn bit to "0".

- 78K0R/KF3-C: P52/TO00, P16/TO01, P17/TO02, P31/TO03, P42/TO04, P05/TO05, P06/TO06, P54/TO07, P64/TO10, P65/TO11, P66/TO12

- 78K0R/KG3-C: P01/TO00, P16/TO01, P17/TO02, P31/TO03, P42/TO04, P46/TO05, P131/TO06, P145/TO07, P64/TO10, P65/TO11, P66/TO12

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 6-19. Format of Timer Output Register m (TOm)

Address: F01B8H, F01B9H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	0	0	TO07	TO06	TO05	TO04	TO03	TO02	TO01	TO00

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	0	TO12	TO11	TO10

TOmn	Timer output of channel n
0	Timer output value is "0".
1	Timer output value is "1".

Caution Be sure to clear bits 15 to 8 of TO0 and bits 15 to 3 of TO1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 12

(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 slave channel output mode (TOMmn = 1). In the master channel output 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 6-20. Format of Timer Output Level Register m (TOLm)

Address: F01BCH, F01BDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	0	0	0	0	TOL07	TOL06	TOL05	TOL04	TOL03	TOL02	TOL01	TOL00

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	0	TOL12	TOL11	TOL10

TOLmn	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 3 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 12

(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 6-21. Format of Timer Output Mode Register m (TOMm)

Address: F01BEH, F01BFH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	0	0	TOM07	TOM06	TOM05	TOM04	TOM03	TOM02	TOM01	TOM00

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	0	TOM12	TOM11	TOM10

TOMmn	Control of timer output mode of channel n
0	Master channel output mode (to produce toggle output by timer interrupt request signal (INTTMmn))
1	Slave channel output mode (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 3 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 2 (n = 0 for master channel)

n < p ≤ 2 (where p is a consecutive integer greater than n)

(13) Noise filter enable registers 1, 2 (NFEN1, NFEN2)

NFEN1 is used to set whether the noise filter can be used for the timer input signal of timer array unit 0 to each channel.

NFEN2 is used to set whether the noise filter can be used for the timer input signal of timer array unit 1 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 of the 2 clocks and synchronization are performed with the CPU/peripheral hardware clock (f_{MCK}). When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock (f_{MCK}).

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 6-22. 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 (TI07/TO07/P54)[Note] pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN06	Enable/disable using noise filter of TI06/TO06/P131 (TI06/TO06/P06)[Note] pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN05	Enable/disable using noise filter of TI05/TO05/INTP1/RIN01/P46 (TI05/TO05/P05)[Note] 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 (TI00/P53) [Note] pin input signal
0	Noise filter OFF
1	Noise filter ON

Note This pin is shared with pin in parentheses in the 78K0R/KF3-C.

Figure 6-23. 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	0	TNFEN12	TNFEN11	TNFEN10

TNFEN12	Enable/disable using noise filter of TI12/TO12/P66 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN11	Enable/disable using noise filter of TI11/TO11/P65 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN10	Enable/disable using noise filter of TI10/TO10/P64 pin input signal
0	Noise filter OFF
1	Noise filter ON

(14) Port mode registers (PMxx)

These registers set input/output of ports in 1-bit units.

When using the following pins for timer output, set the port mode register (PMxx) bit and the port register (Pxx) bit corresponding to each port to 0.

- 78K0R/KF3-C: P52/TO00, P16/TO01/TI01/INTP5, P17/TO02/TI02, P31/TO03/TI03/INTP4, P42/TO04/TI04, P05/TO05/TI05, P06/TO06/TI06, P54/TO07/TI07, P64/TI10/TO10 to P66/TI12/TO12

- 78K0R/KG3-C: P01/TO00, P16/TO01/TI01/INTP5, P17/TO02/TI02, P31/TO03/TI03/INTP4, P42/TO04/TI04, P46/TO05/TI05/INTP1/RIN01, P131/TO06/TI06, P145/TO07/TI07, P64/TI10/TO10 to P66/TI12/TO12

Example: When using P16/TO01/TI01/INTP5 for timer output
Set the PM16 bit of port mode register 1 to 0.
Set the P16 bit of port register 1 to 0.

When using the following pins for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

- 78K0R/KF3-C: P53/TI00, P16/TO01/TI01/INTP5, P17/TO02/TI02, P31/TO03/TI03/INTP4, P42/TO04/TI04, P05/TO05/TI05, P06/TO06/TI06, P54/TO07/TI07, P64/TI10/TO10 to P66/TI12/TO12

- 78K0R/KG3-C: P00/TI00, P16/TO01/TI01/INTP5, P17/TO02/TI02, P31/TO03/TI03/INTP4, P42/TO04/TI04, P46/TO05/TI05/INTP1/RIN01, P131/TO06/TI06, P145/TO07/TI07, P64/TI10/TO10 to P66/TI12/TO12

Example: When using P16/TO01/TI01/INTP5 for timer input
Set the PM16 bit of port mode register 1 to 1.
Set the P16 bit of port register 1 to 0 or 1.

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

Reset signal generation sets these registers to FFH.

Figure 6-24. Format of Port Mode Registers (78K0R/KF3-C)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	1	1

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	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: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50

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

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3 to 6; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 6-25. Format of Port Mode Registers (78K0R/KG3-C)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	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	1	1	1	1	1	1	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: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60

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

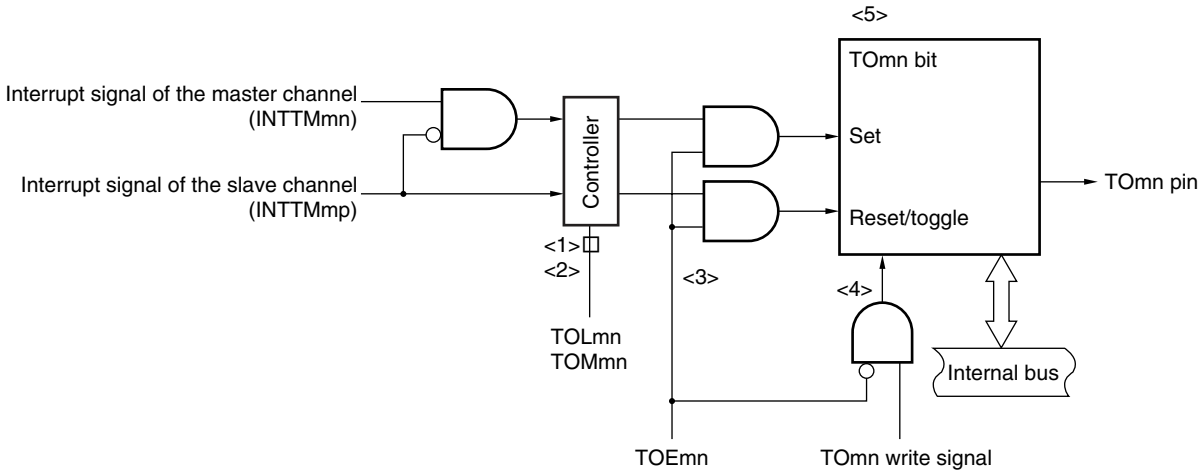
Symbol	7	6	5	4	3	2	1	0
PM14	1	1	PM145	PM144	PM143	PM142	PM141	PM140

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 4, 6, 13, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 Channel Output (TOmn pin) Control

6.4.1 TOmn pin output circuit configuration

Figure 6-26. Output Circuit Configuration



The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (master channel output mode), the set value of the TOLmn bit is ignored and only INTTMmp (slave channel timer interrupt) is transmitted to the TOmn bit.

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are transmitted to the TOmn bit.
At this time, the TOLmn bit 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 bit. Writing to the TOmn bit (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 bit.

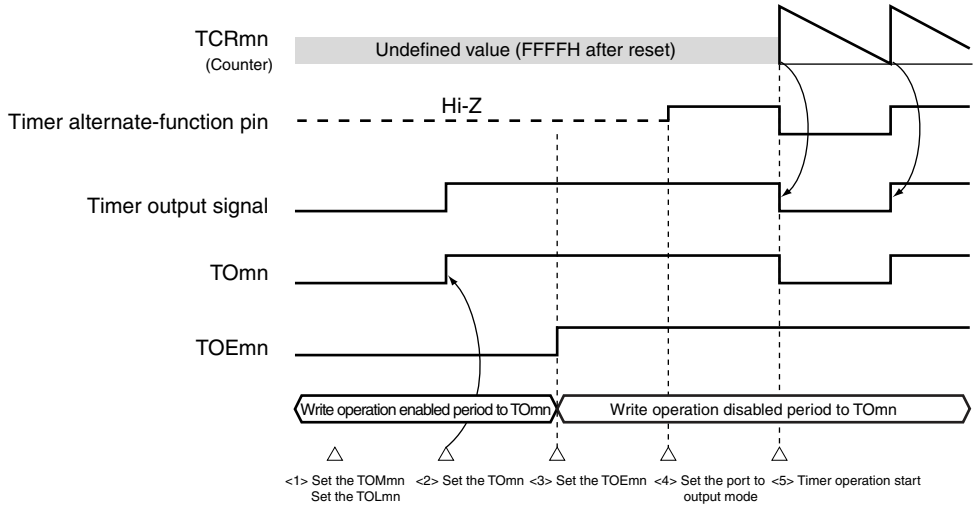
<5> The TOmn bit can always be read, and the TOmn pin output level can be checked.

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 2 (n = 0 for master channel)
n < p ≤ 2 (where p is a consecutive integer greater than n)

6.4.2 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 6-27. Status Transition from Timer Output Setting to Operation Start



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

- TOMmn bit (0: Master channel output mode, 1: Slave channel output mode)
- TOLmn bit (0: 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 **6.3 (14) Port mode registers 0, 1, 3, 4, 6, 13, 14**).
<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 12

6.4.3 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 12

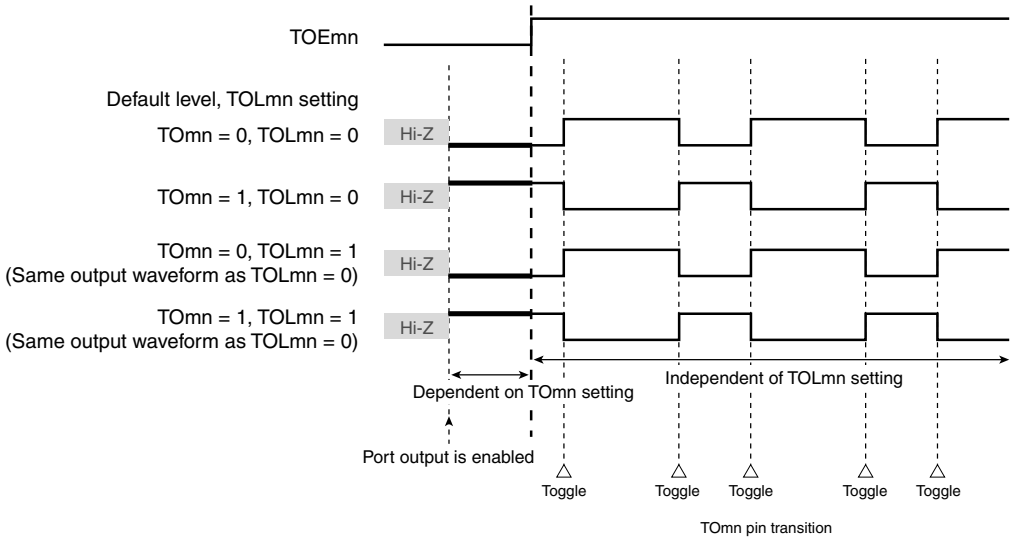
(2) Default level of TOmn pin and output level after timer operation start

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 6-28. TOmn Pin Output Status at Toggle Output (TOMmn = 0)



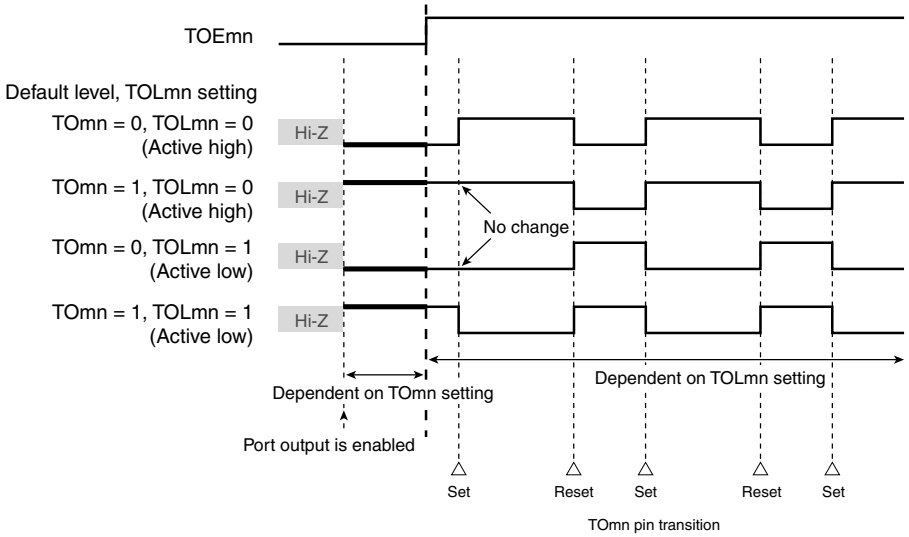
Remarks 1. Toggle: Reverse TOmn pin output status

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

(b) When operation starts with TOMmn = 1 setting (Slave channel output mode (PWM output))

When TOMmn = 1, the active level is determined by TOLmn setting.

Figure 6-29. 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 12

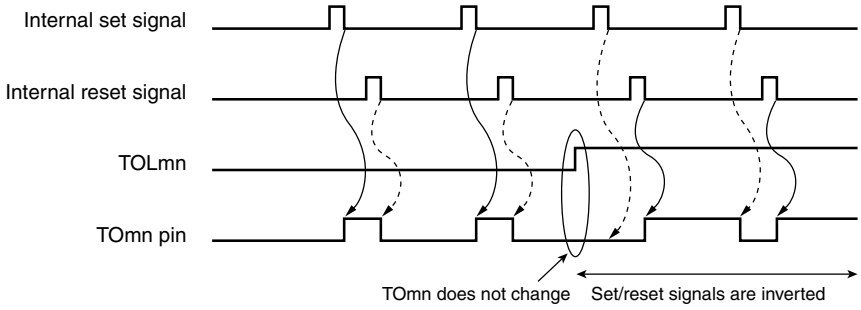
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(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 6-30. 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 12

(b) Set/reset timing

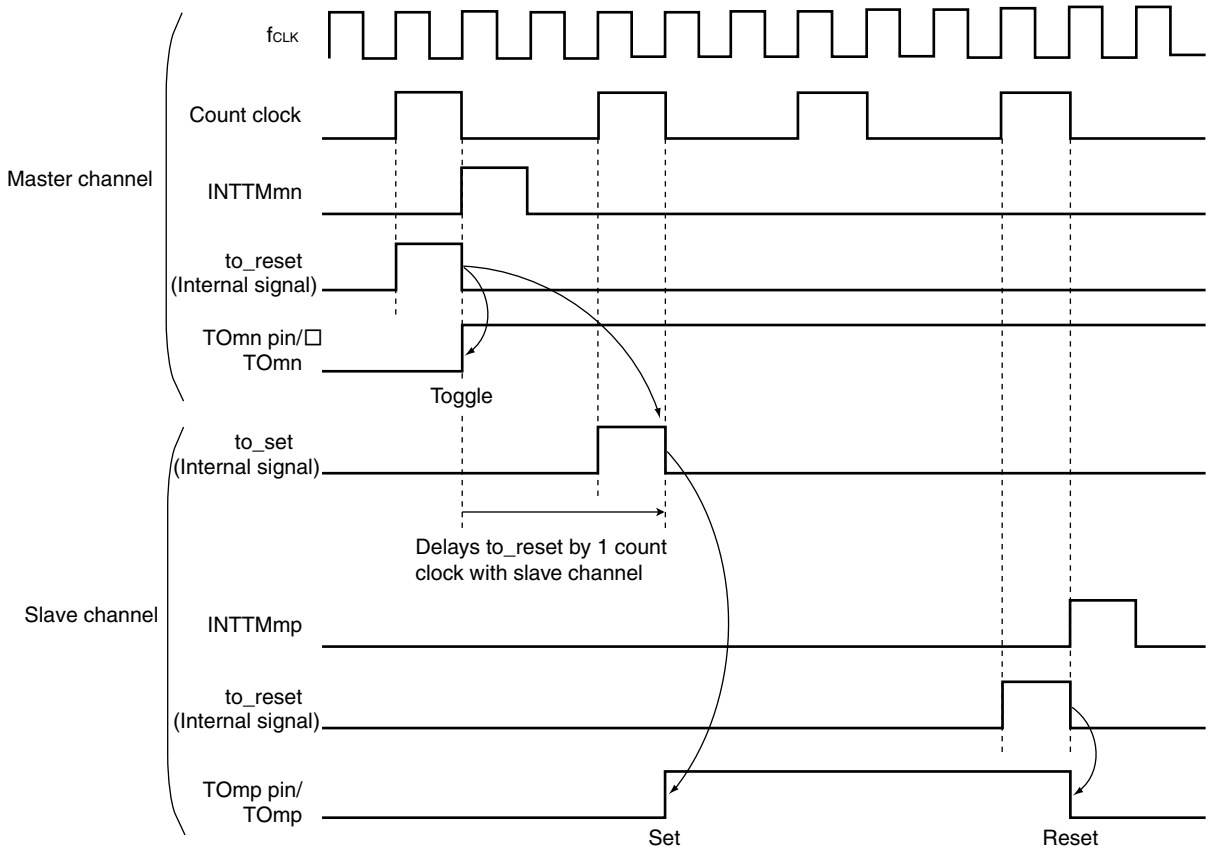
To realize 0%/100% output at PWM output, the TOmn pin/TOmn set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

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

Figure 6-31 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 6-31. Set/Reset Timing Operating Statuses



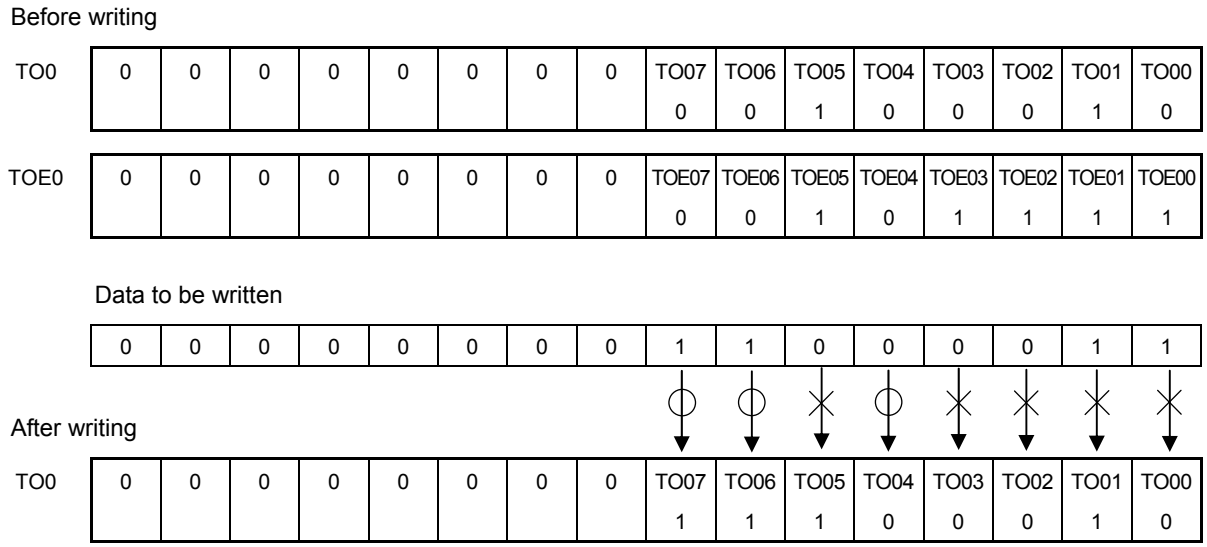
Remarks 1. to_reset: TOmn pin reset/toggle signal
to_set: TOmn pin set signal

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 2 (n = 0 for master channel)
n < p ≤ 2 (where p is a consecutive integer greater than n)

6.4.4 Collective manipulation of TOmn bits

In the TOm 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, TOmn of all the channels can be manipulated collectively. Only specific bits can also be manipulated by setting the corresponding TOEmn = 0 to a target TOmn (channel output).

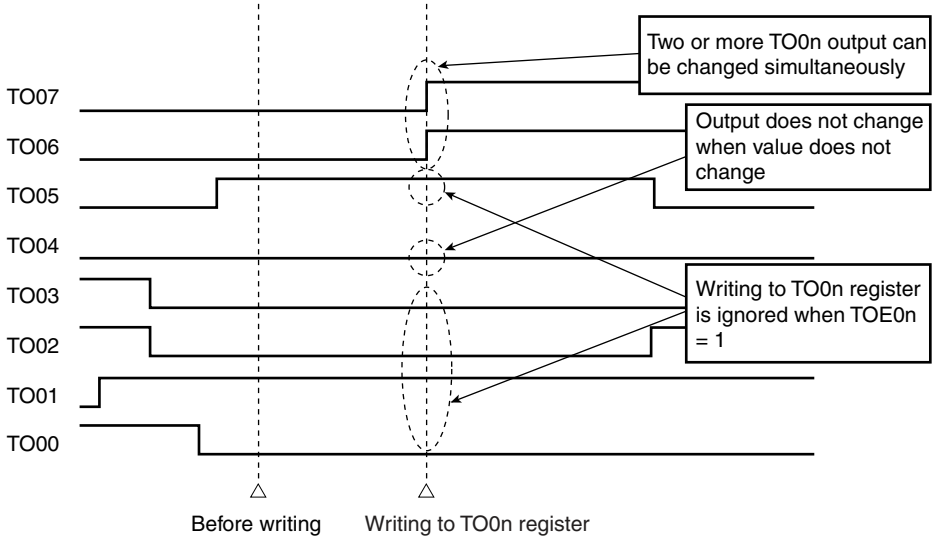
Figure 6-32. Example of TO0n Bits Collective Manipulation



Writing is done only to TOmn bits with TOEmn = 0, and writing to TOmn bits with TOEmn = 1 is ignored.

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

Figure 6-33. TO0n Pin Statuses by Collective Manipulation of TO0n Bits



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 12

6.4.5 Timer Interrupt and TOmn Pin Output at Operation Start

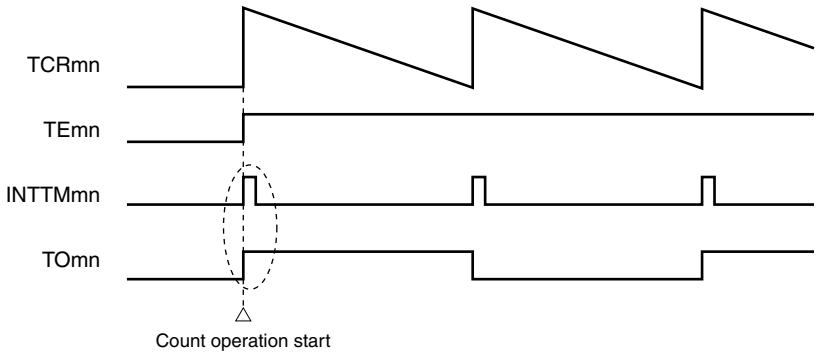
In the interval timer mode or capture mode, the MDmn0 bit in the TMRmn register sets whether or not to generate a timer interrupt at count start.

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.

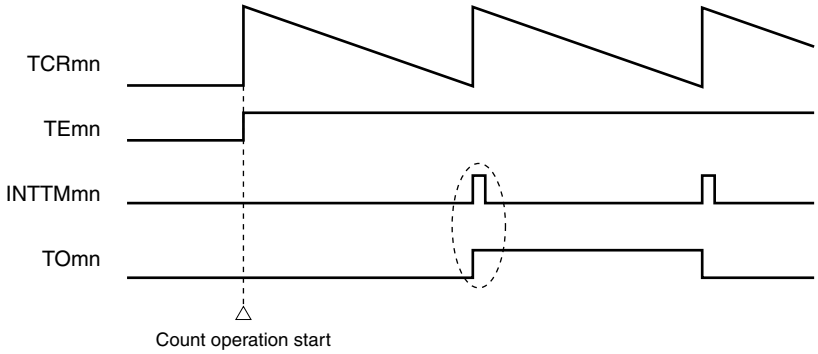
Figures 6-34 and 6-35 show operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 6-34. 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 6-35. 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 12

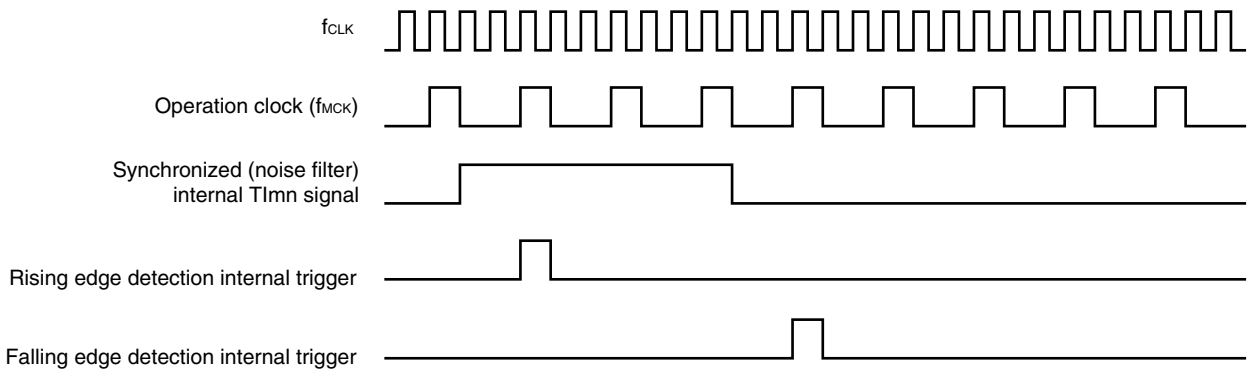
6.5 Channel Input (TImn Pin) Control

6.5.1 TImn edge detection circuit

(1) Edge detection basic operation timing

Edge detection circuit sampling is done in accordance with the operation clock (f_{MCK}).

Figure 6-36. 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 12

6.6 Basic Function of Timer Array Unit

6.6.1 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.

6.6.2 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 0 of the TAU1 is set as a master channel, channel 1 or 2 (only up to channel2 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 higher 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.

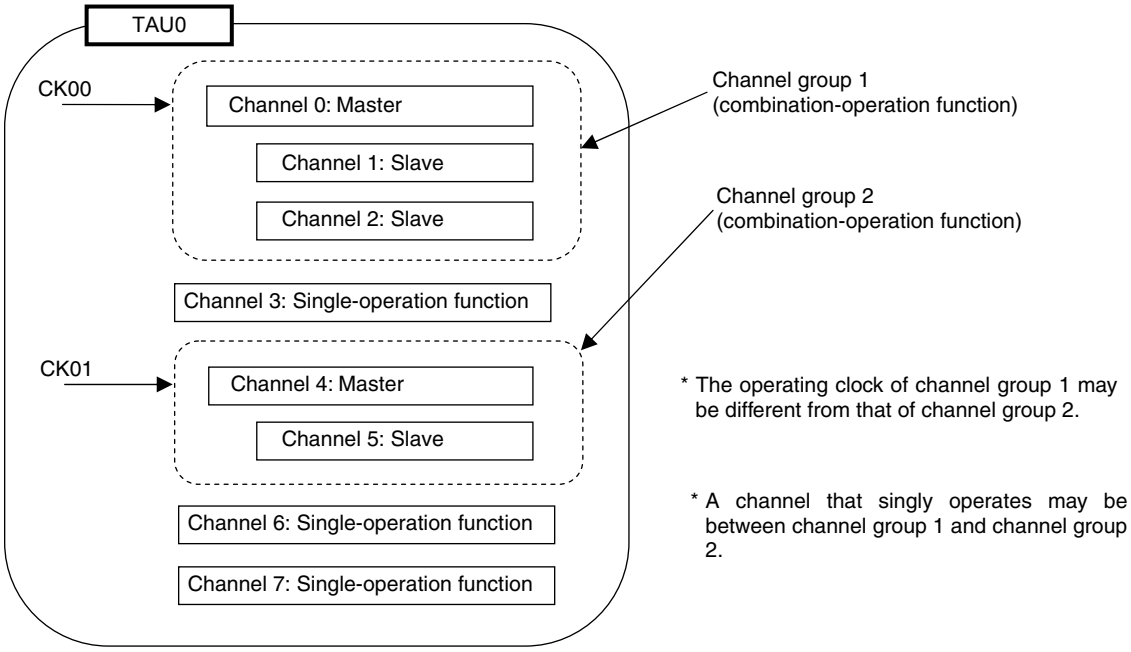
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

6.6.3 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 **6.6.2 Basic rules of combination-operation function** do not apply to the channel groups.

Example



6.7 Operation of Timer Array Unit as Independent Channel

6.7.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTMmn (timer interrupt) at fixed intervals. The interrupt generation period can be calculated by the following expression.

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

A subsystem clock divided by four ($f_{SUB}/4$) 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 $f_{SUB}/4$, regardless of the f_{CLK} frequency (main system clock, subsystem clock). When changing the clock selected as f_{CLK} (changing the value of the system clock control register (CKC)), however, stop the timer array unit (TAU) (TT0 = 00FFH, TT1 = 0007H) 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.

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

TCRmn operates as a down counter in the interval timer mode.

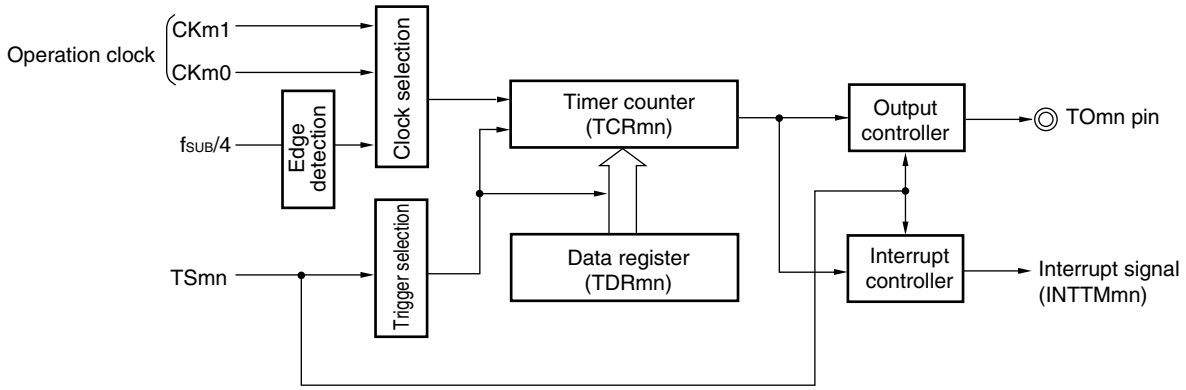
TCRmn loads the value of TDRmn at the first count clock after the channel start trigger bit (TSmn) is set to 1. If MDmn0 of TMRmn = 0 at this time, INTTMmn is not output and TOmn is not toggled. If MDmn0 of TMRmn = 1, INTTMmn is output and TOmn is toggled.

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.

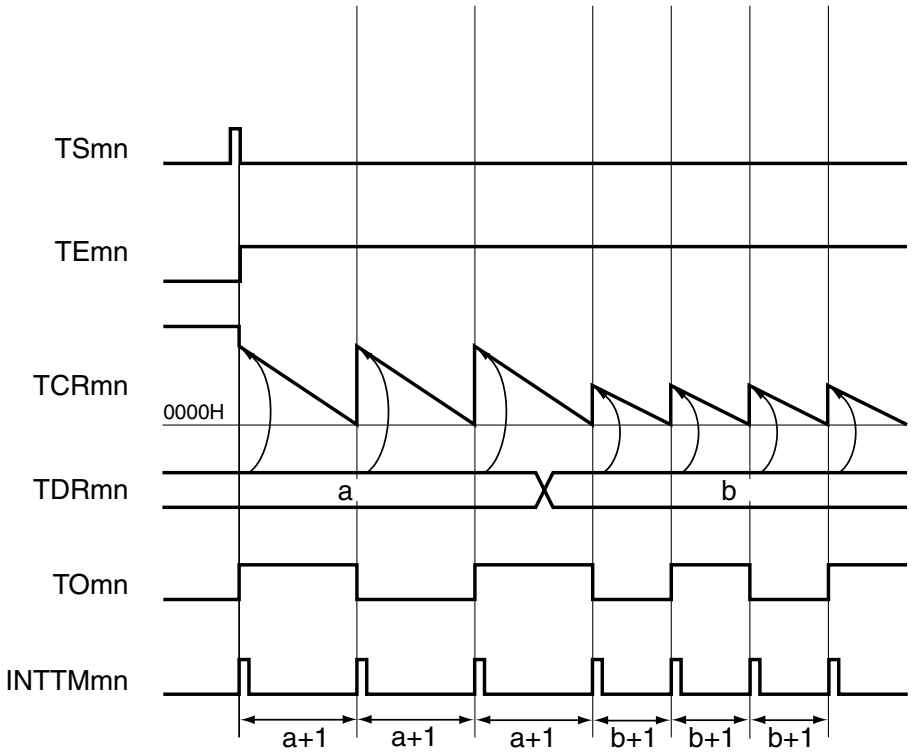
TDRmn can be rewritten at any time. The new value of TDRmn becomes valid from the next period.

Figure 6-37. Block Diagram of Operation as Interval Timer/Square Wave Output



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

Figure 6-38. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

Figure 6-39. 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)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn			CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1/0

Operation mode of channel n
000B: Interval timer

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

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

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CKm0 as operation clock of channel n.
1: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	1/0

0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	1/0

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

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 12

Figure 6-39. 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 (1/2)

(a) Timer mode register mn (TMRmn)

TMRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKSmn			CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	0	0	1/0

Operation mode of channel n
000B: Interval timer

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

f_{SUB}/4 edge selection
00B: Detects falling edge (counts on f_{SUB}/4 cycles).
01B: Detects rising edge (counts on f_{SUB}/4 cycles).
10B: Detects both edges (counts on f_{SUB}/2 cycles).
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
1: Selects subsystem clock divided by four (f_{SUB}/4).

Operation clock selection
0: Selects CKm0 as operation clock of channel n.
1: Selects CKm1 as operation clock of channel n.
f_{CLK} (no division) is selected as selected operation clock by TPSm register.

(b) Timer clock select register m (TPSm)

TPSm

Bits 7 to 4, 3 to 0
PRSmk3 to PRSmk0
0000

0000B: Selects f_{CLK} (no division) as operation clock selected by CKSmn of TMRmn register.
k = 0 (bits 0 to 3) when CKm0 is selected and k = 1 (bits 4 to 7) when CKm1 is selected

(c) Timer input select register m (TISm)

TISm

Bit n
TISmn
1

1: Selects subsystem clock divided by four (f_{SUB}/4).

(d) Timer output register m (TOm)

TOm

Bit n
TOmn
1/0

0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 12

2. f_{SUB}: Subsystem clock oscillation frequency

Figure 6-39. 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 (2/2)

(e) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 1/0	0: Stops the TOmn output operation by counting operation. 1: Enables the TOmn output operation by counting operation.

(f) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(g) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7), mn = 00 to 07, 10 to 12

2. f_{SUB}: Subsystem clock oscillation frequency

Figure 6-40. 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 (master channel output mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output. →	The TOmn pin goes into Hi-Z output state. The TOmn default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets TOEmn to 1 and enables operation of TOmn. →	TOmn does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmn pin outputs the TOmn set level.
Operation start	(Sets the TOEmn bit to 1 only if using TOmn output and resuming operation.). The TTmn bit is set 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 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 TOm register. →	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 12

Figure 6-40. 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 12

6.7.2 Operation as external event counter

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

Specified number of counts = Set value of TDRmn + 1

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.

An irregular waveform that depends on external events is output from the TOmn pin. Stop the output by setting the TOEmn bit of timer output enable register m (TOEm) to 0.

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

Figure 6-41. Block Diagram of Operation as External Event Counter

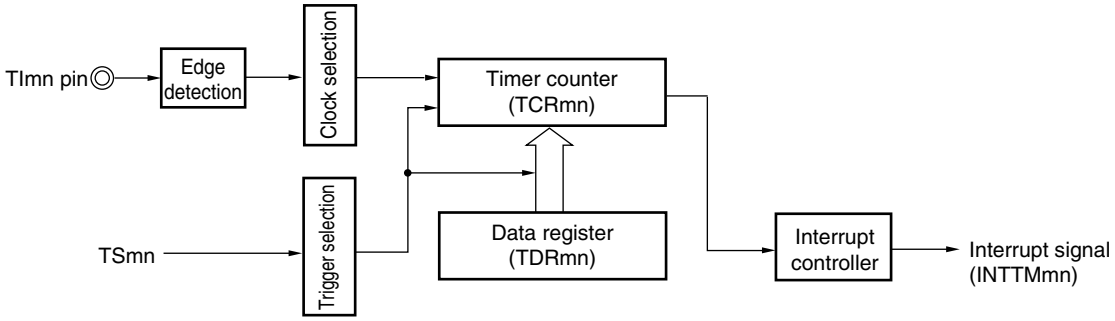
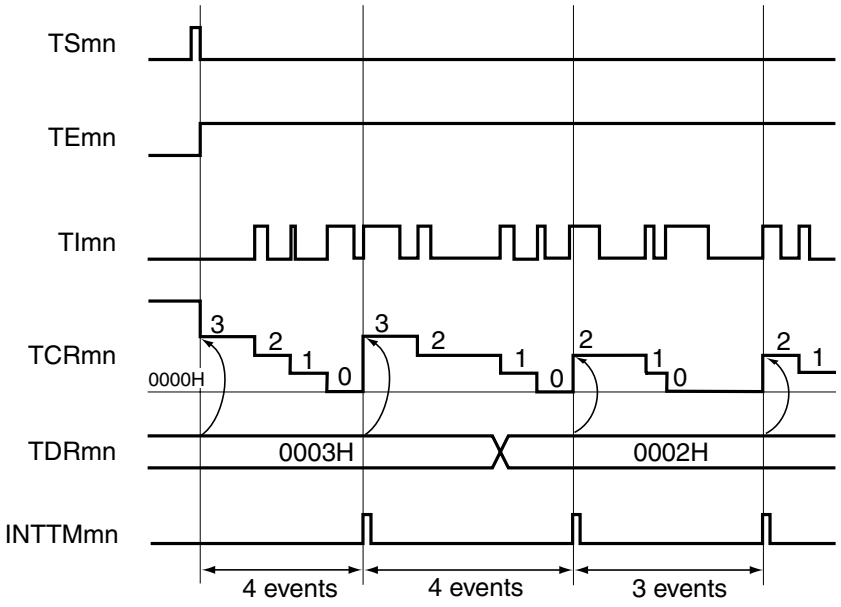


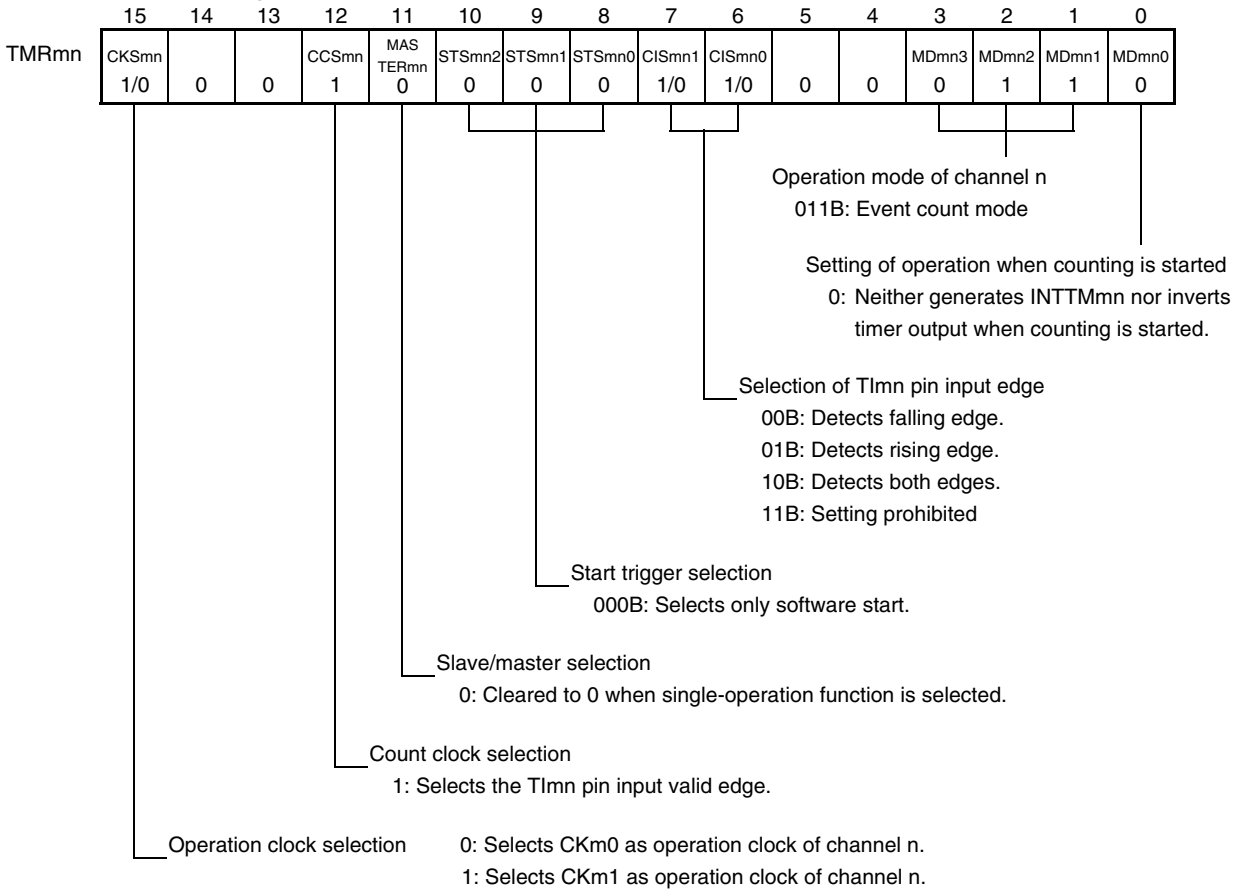
Figure 6-42. Example of Basic Timing of Operation as External Event Counter



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

Figure 6-43. Example of Set Contents of Registers in External Event Counter Mode

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

Figure 6-44. 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 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 12

6.7.3 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.

Clock period of TO00 output = Ideal TO00 output clock period ± Operation clock period (error)

TDR00 can be rewritten at any time. The new value of TDR00 becomes valid during the next count period.

Figure 6-45. Block Diagram of Operation as Frequency Divider

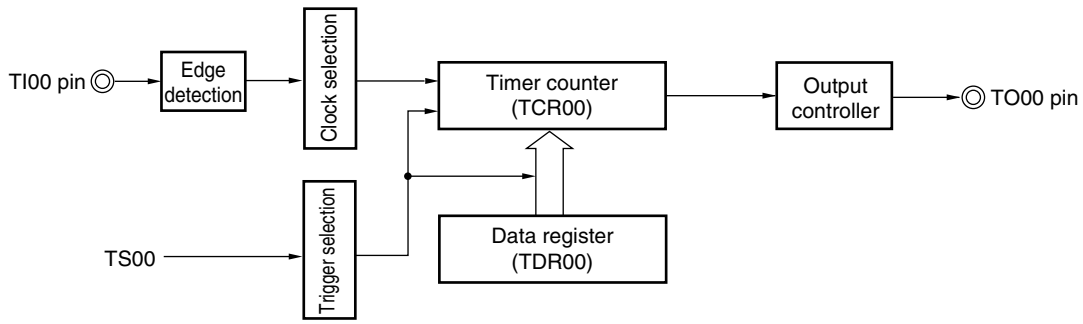


Figure 6-46. Example of Basic Timing of Operation as Frequency Divider (MD000 = 1)

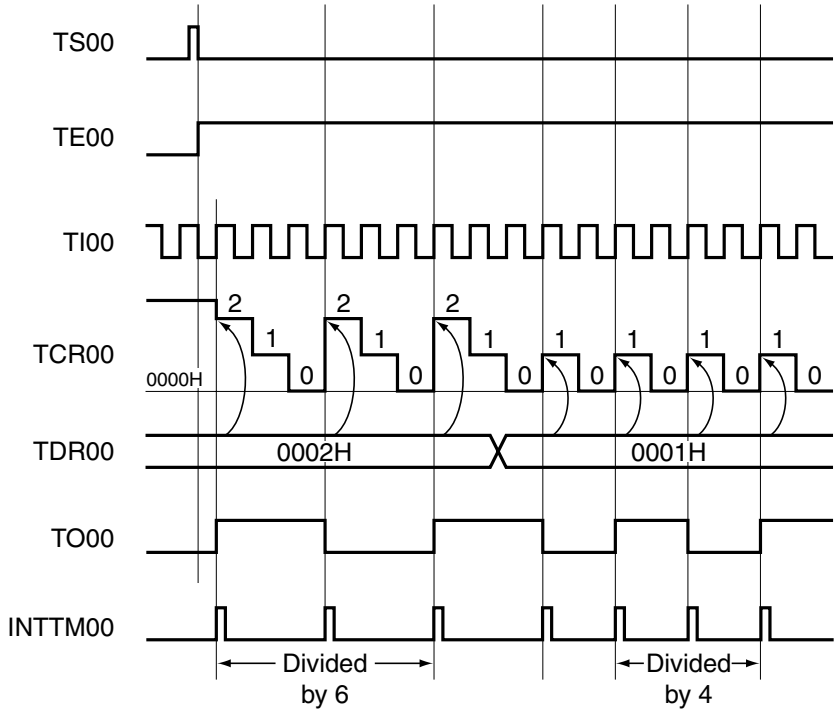
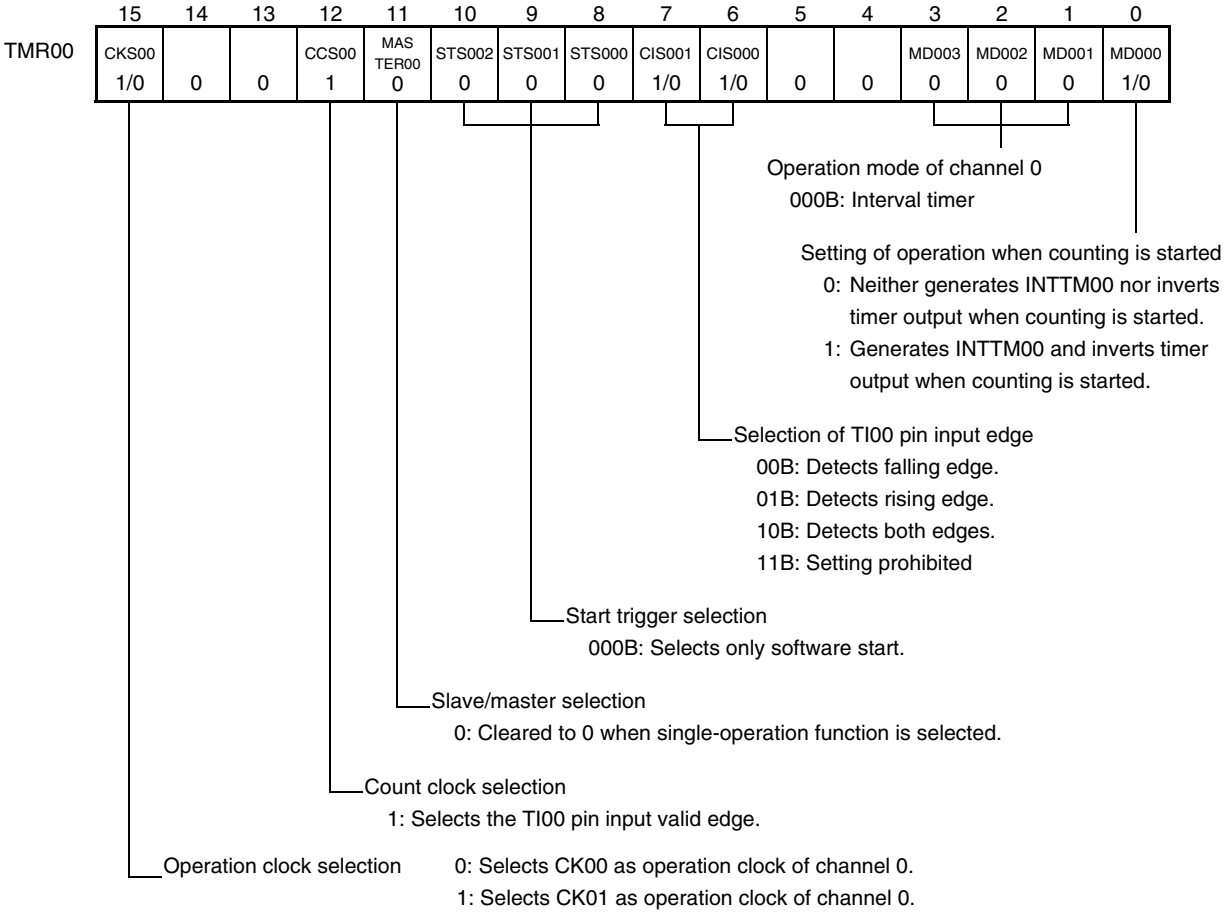


Figure 6-47. Example of Set Contents of Registers When Frequency Divider Is Used

(a) Timer mode register 00 (TMR00)



(b) Timer output register 0 (TO0)

	Bit 0	
TO0	TO00 1/0	0: Outputs 0 from TO00. 1: Outputs 1 from TO00.

(c) Timer output enable register 0 (TOE0)

	Bit 0	
TOE0	TOE00 1/0	0: Stops the TO00 output operation by counting operation. 1: Enables the TO00 output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit 0	
TOL0	TOL00 0	0: Cleared to 0 when TOM00 = 0 (master channel output mode)

(e) Timer output mode register 0 (TOM0)

	Bit 0	
TOM0	TOM00 0	0: Sets master channel output mode.

Figure 6-48. 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, 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 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 (master channel output 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. →	TO00 does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO00 pin outputs the TO00 set level.
Operation start	Sets the TOE00 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 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 TO0 register. →	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. → When holding the TO00 pin output level is not necessary Switches the port mode register to input mode. →	The TO00 pin output level is held by port function. The TO00 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 TO00 bit is cleared to 0 and the TO00 pin is set to port mode).

Operation is resumed.

6.7.4 Operation as input pulse interval measurement

The count value can be captured at the 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.

TImn input pulse interval = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of 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.

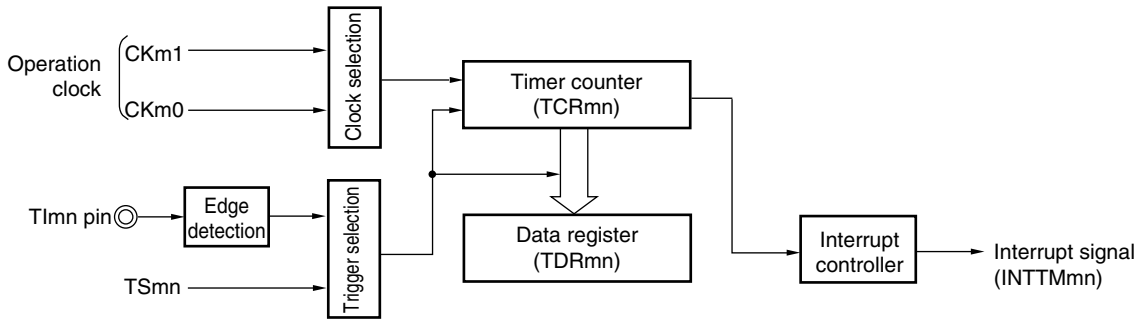
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

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.

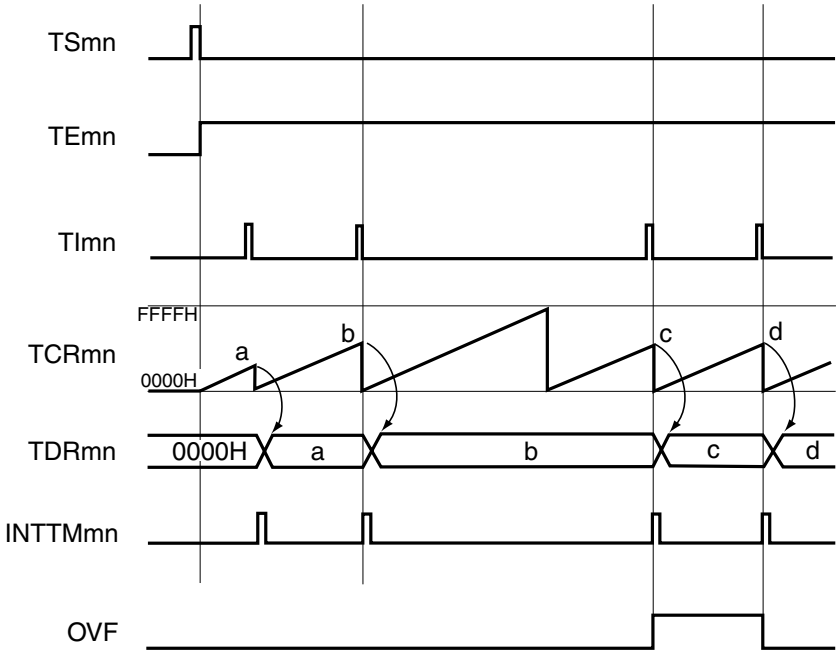
When TEmn = 1, instead of the TImn pin input, a software operation (TSmn = 1) can be used as a capture trigger.

Figure 6-49. 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 12

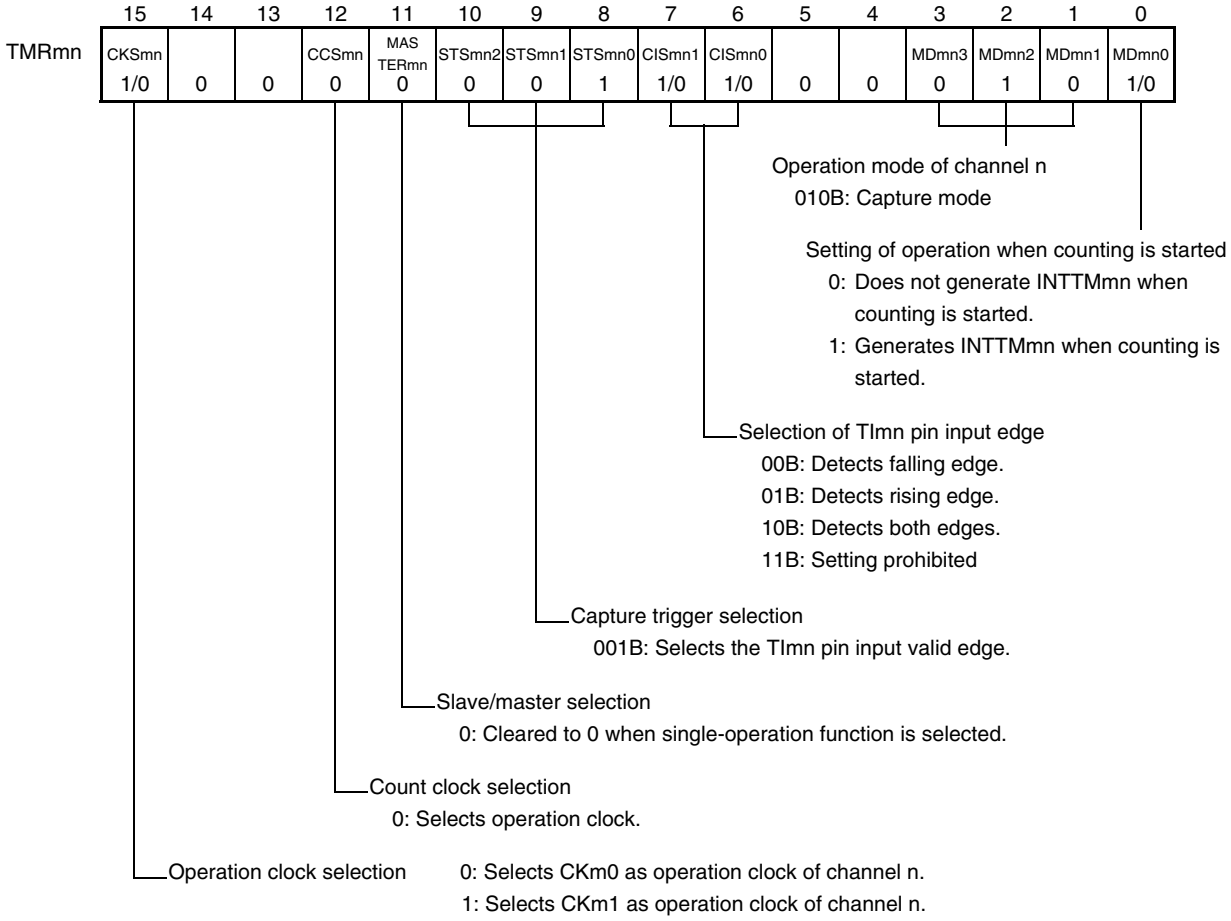
Figure 6-50. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

Figure 6-51. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

Figure 6-52. 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. →	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 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. →	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. ▶ (Operation start)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

6.7.5 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.

Signal width of TImn input = Period of count clock × ((10000H × TSRn: OVF) + (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 equivalent to one operation clock 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.

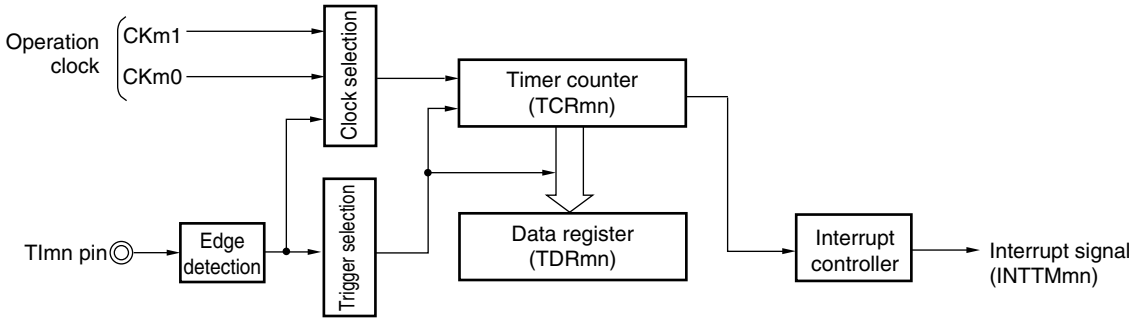
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.

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.

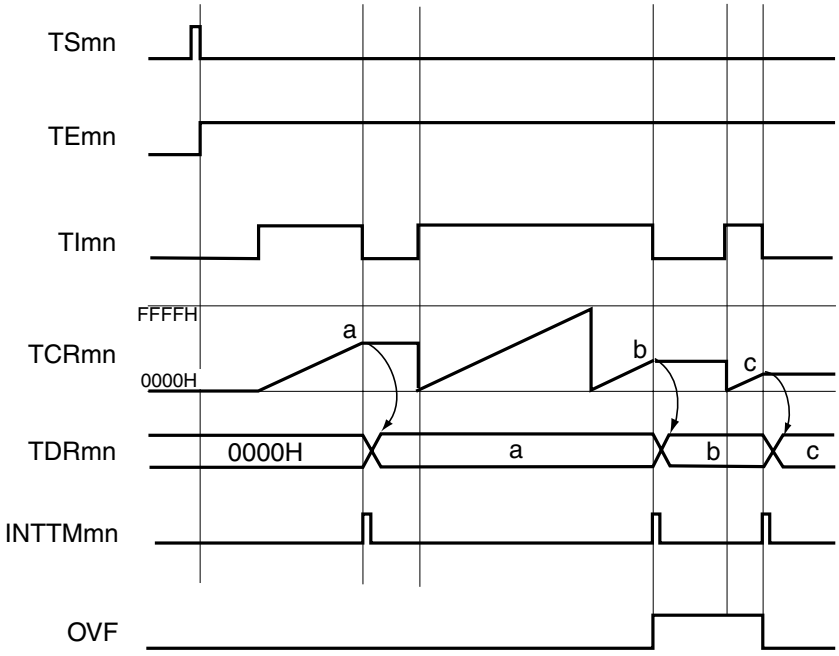
CISmn1, CISmn0 of TMRmn = 11B: High-level width is measured.

Figure 6-53. 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 12

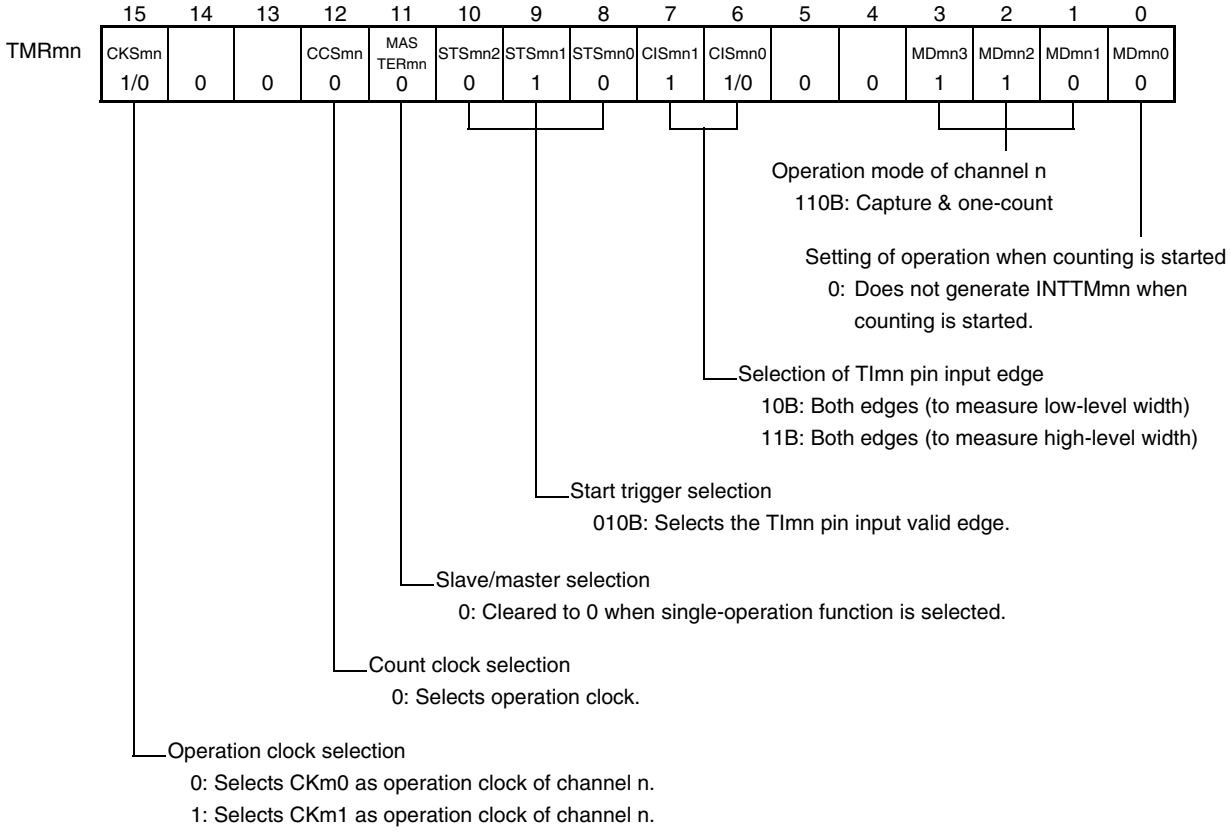
Figure 6-54. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



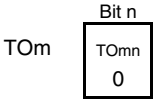
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

Figure 6-55. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register mn (TMRmn)

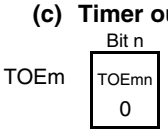


(b) Timer output register m (TOm)



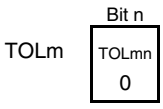
0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)



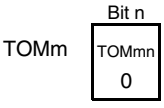
0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)



0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)



0: Sets master channel output mode.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7),
mn = 00 to 07, 10 to 12

Figure 6-56. 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 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 12

6.8 Operation of Plural Channels of Timer Array Unit

6.8.1 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.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDRmp (slave)}/{Set value of TDRmn (master) + 1} × 100
0% output: Set value of TDRmp (slave) = 0000H
100% output: Set value of TDRmp (slave) ≥ {Set value of TDRmn (master) + 1}

Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, an interrupt (INTTMmn) is output, the value set to timer data register mn (TDRmn) is loaded to timer/counter register mn (TCRmn), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTMmn is output, the value of the TDRmn register is loaded again to the TCRmn register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TTmn) of timer channel stop register m (TTm) is set to 1.

If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TOmp) cycle.

The slave channel operates in one-count mode. By using INTTMmn from the master channel as a start trigger, the TCRmp register loads the value of the TDRmp register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTMmp and waits until the next start trigger (INTTMmn from the master channel) is generated.

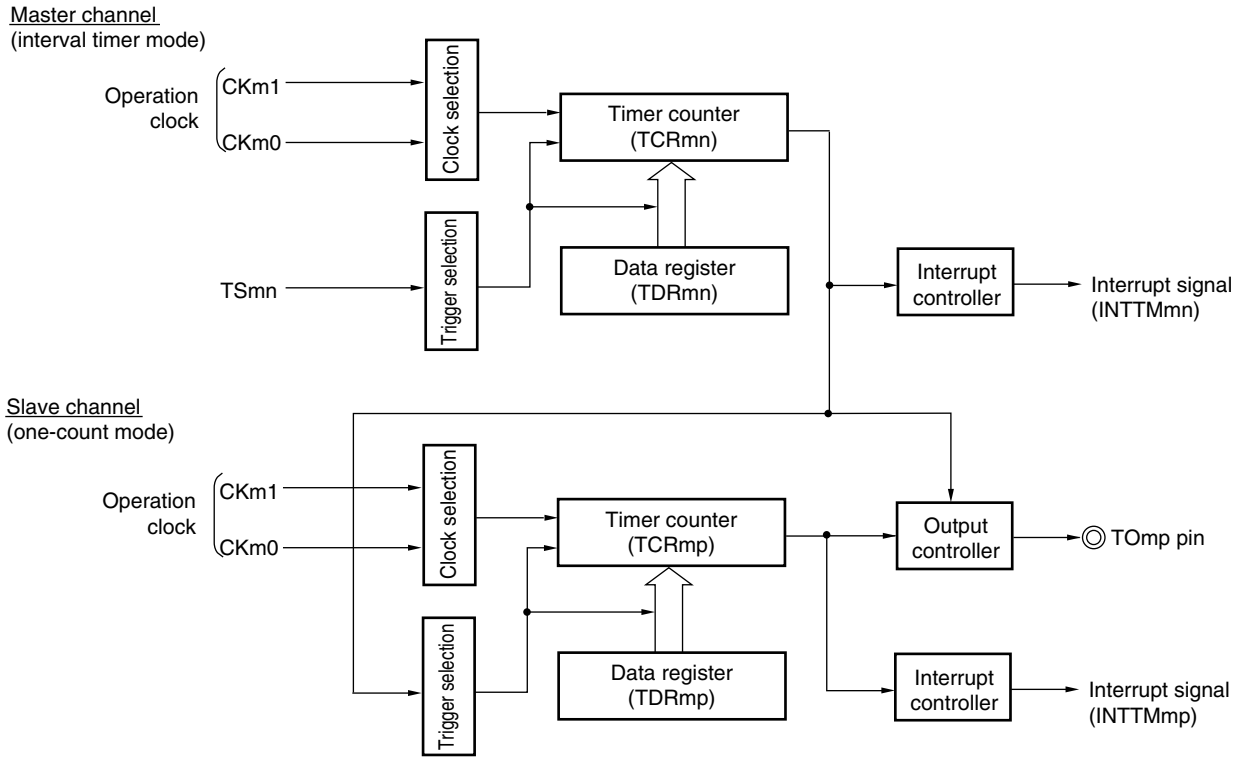
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.

PWM output (TOmp) goes to the active level one clock after the master channel generates INTTMmn and goes to the inactive level when the TCRmp register of the slave channel becomes 0000H.

Caution To rewrite both 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 TCRmp 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.

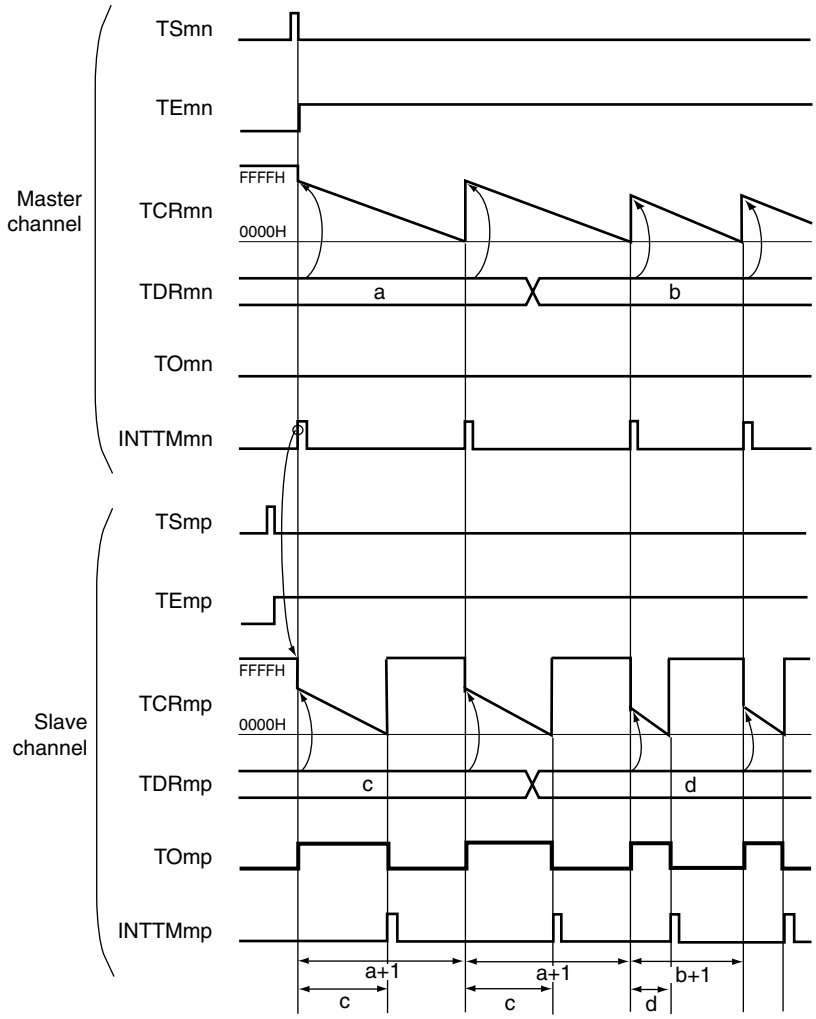
Remark m: Unit number, n: Channel number, p: Slave channel number (p = n+1)
When m = 0: n = 0, 2, 4, 6
When m = 1: n = 0

Figure 6-57. 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

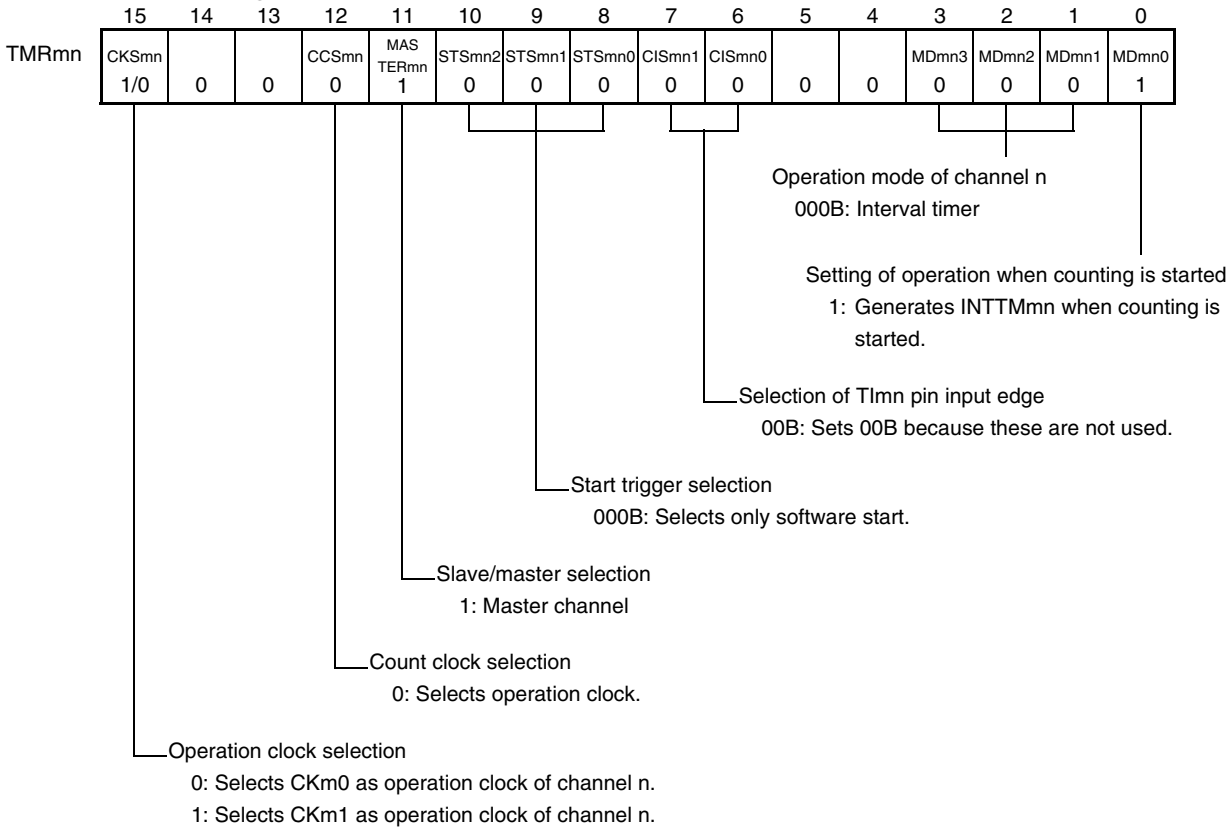
Figure 6-58. Example of Basic Timing of Operation as PWM Function



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

Figure 6-59. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn	0: Outputs 0 from TOmn.
	0	

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn	0: Stops the TOmn output operation by counting operation.
	0	

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn	0: Cleared to 0 when TOMmn = 0 (master channel output mode).
	0	

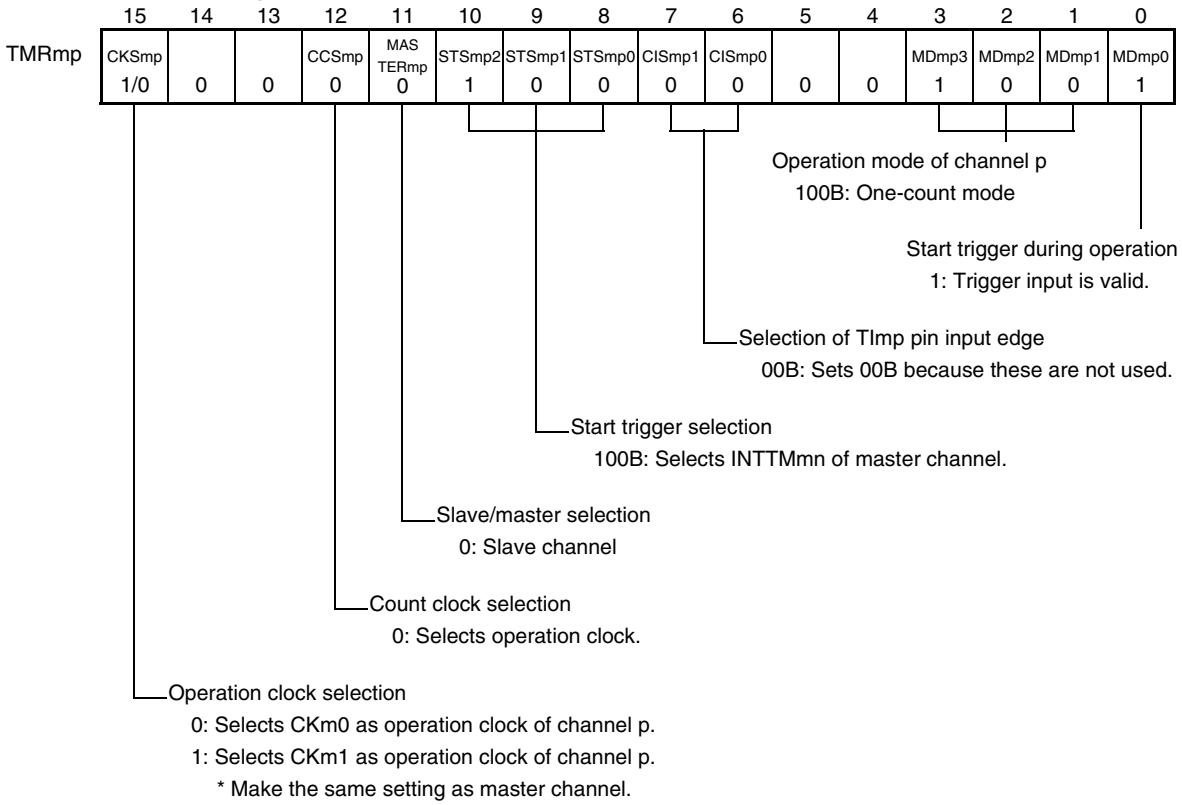
(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn	0: Sets master channel output mode.
	0	

Remark m: Unit number, n: Channel number
When m = 0: n = 0, 2, 4, 6
When m = 1: n = 0

Figure 6-60. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register mp (TMRmp)



(b) Timer output register m (TOm)

	Bit p	
TOm	TOmp 1/0	0: Outputs 0 from TOmp. 1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p	
TOEm	TOEmp 1/0	0: Stops the TOmp output operation by counting operation. 1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p	
TOLm	TOLmp 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p	
TOMm	TOMmp 1	1: Sets slave channel output mode.

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

Figure 6-61. 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 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEmp to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

Remark 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

Figure 6-61. Operation Procedure When PWM Function Is Used (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 = 1, TEmp = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
	The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	
During operation	Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers cannot be changed.	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.
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, TAU1ENbit 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

6.8.2 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.

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

The Master channel operates in the one-count mode and counts the delays. 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.

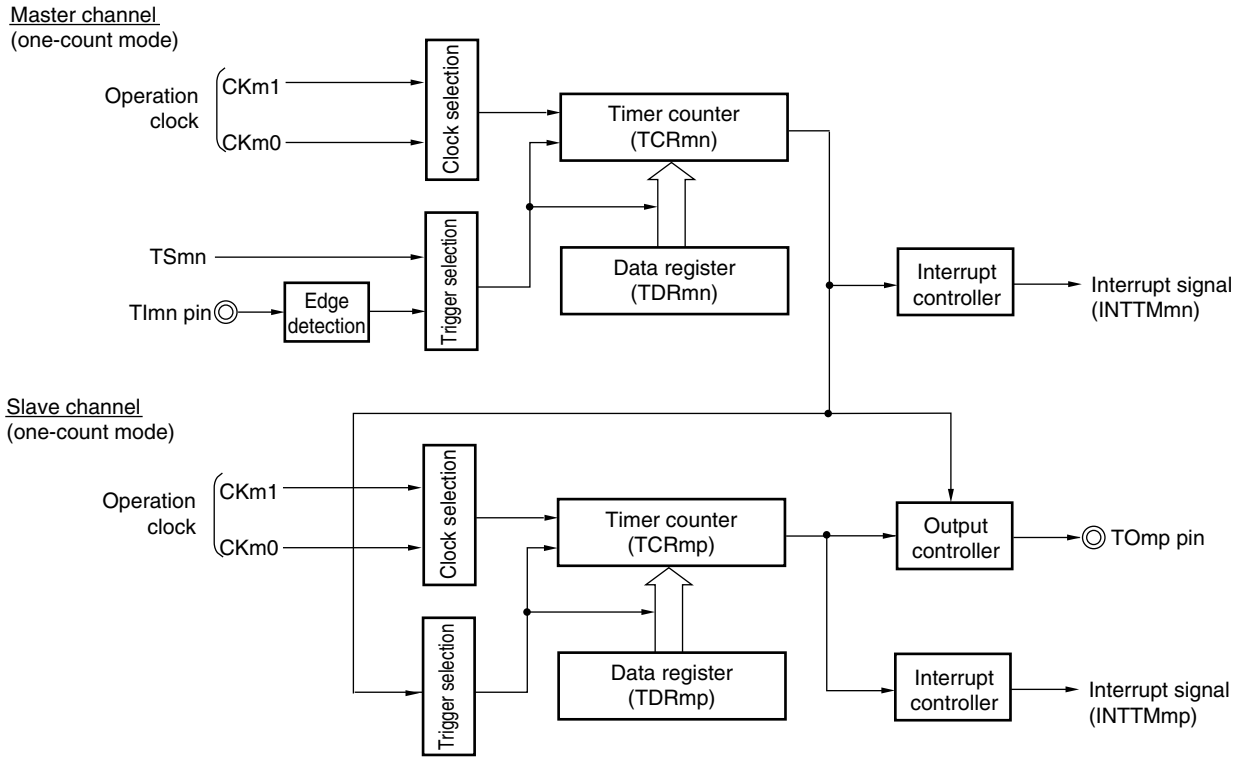
The slave channel operates in the one-count mode and counts the pulse width. TCRmp of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the TDRmp value. TCRmp counts down from the value of TDRmp it has loaded, in synchronization with the count value. 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.

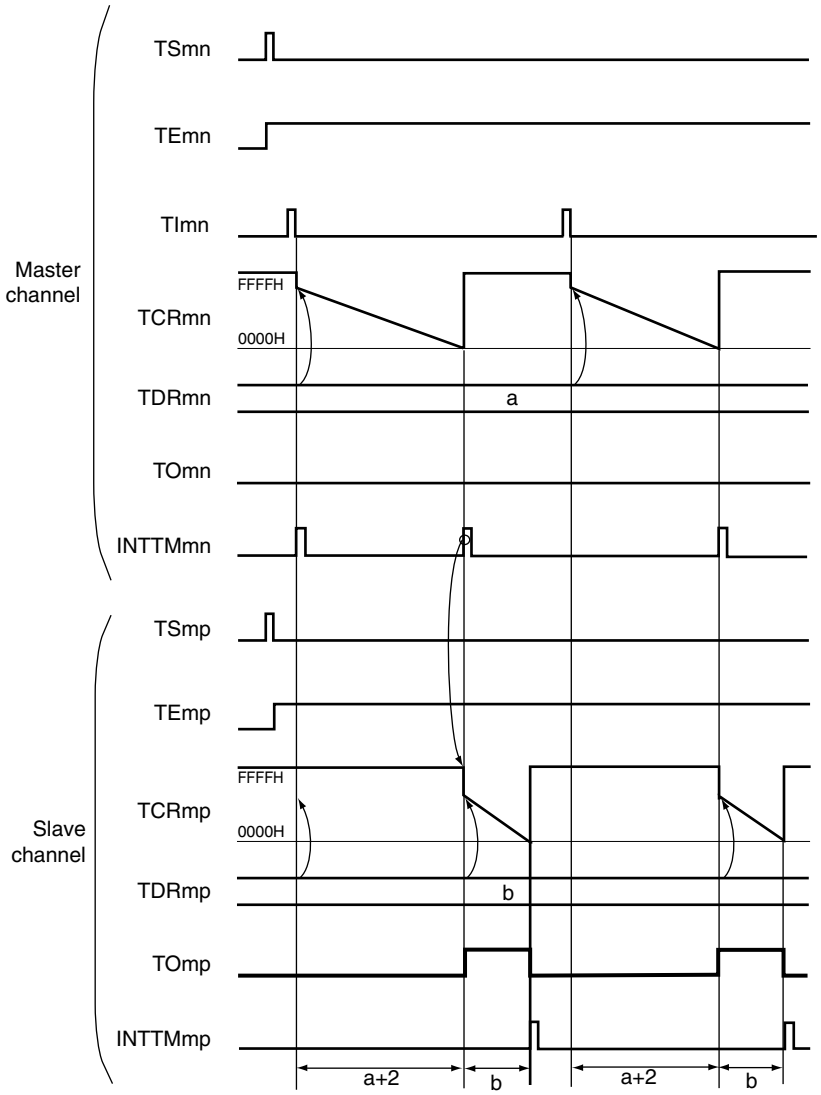
Remark m: Unit number, n: Channel number, p: Slave channel number (p = n+1)
When m = 0: n = 0, 2, 4, 6
When m = 1: n = 0

Figure 6-62. 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

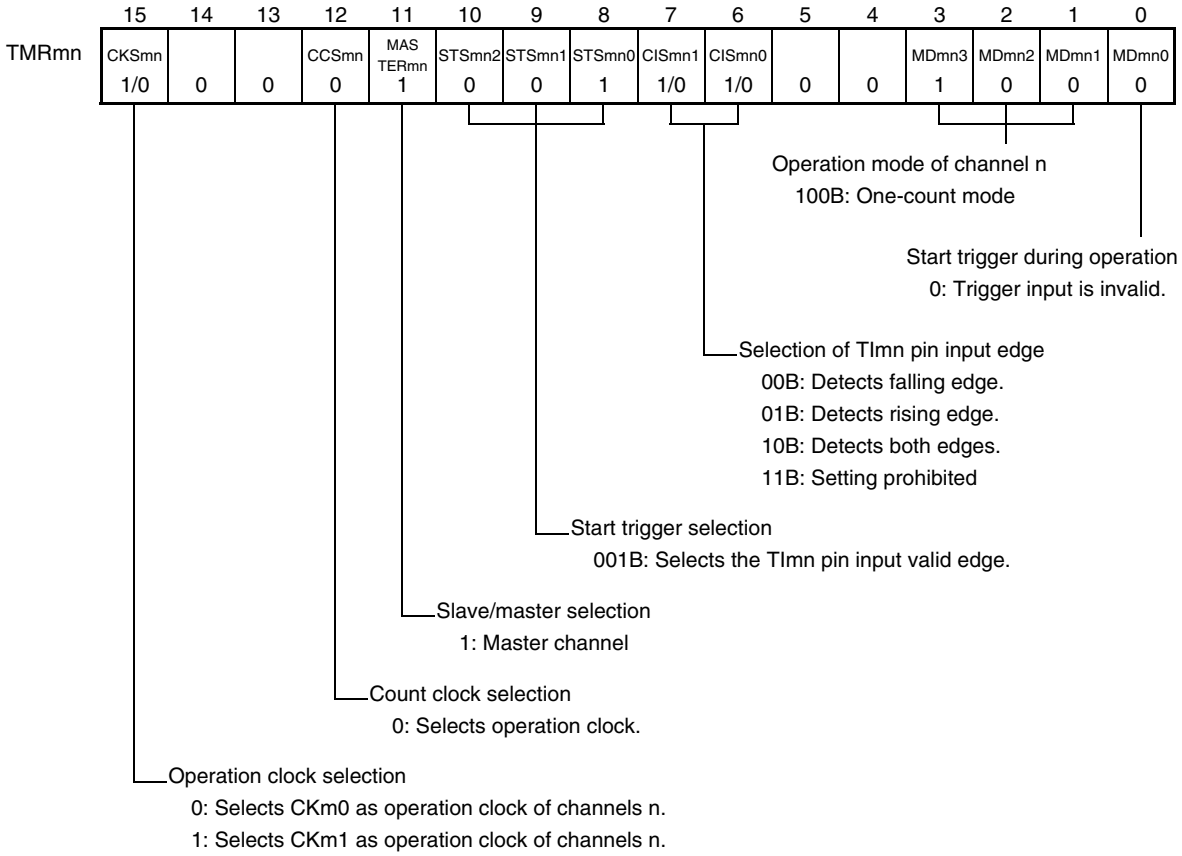
Figure 6-63. Example of Basic Timing of Operation as One-Shot Pulse Output Function



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

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

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

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

Figure 6-65. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register mp (TMRmp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp			CCSmp	MAS TERmp	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

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

Start trigger during operation
0: Trigger input is invalid.

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

Start trigger selection
100B: Selects INTTMmn of master channel.

Slave/master selection
0: Slave channel

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CKm0 as operation clock of channel p.
1: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p
TOm	TOmp
	1/0

0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p
TOEm	TOEmp
	1/0

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

(d) Timer output level register m (TOLm)

	Bit p
TOLm	TOLmp
	1/0

0: Positive logic output (active-high)
1: Inverted output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p
TOMm	TOMmp
	1

1: Sets slave channel output mode.

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

Figure 6-66. Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit, 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 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmn pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOEmp to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

Remark 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

Figure 6-66. 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.
	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.
	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

6.8.3 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.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDRmp (slave 1)}/{Set value of TDRmn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRmq (slave 2)}/{Set value of TDRmn (master) + 1} × 100

Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

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.

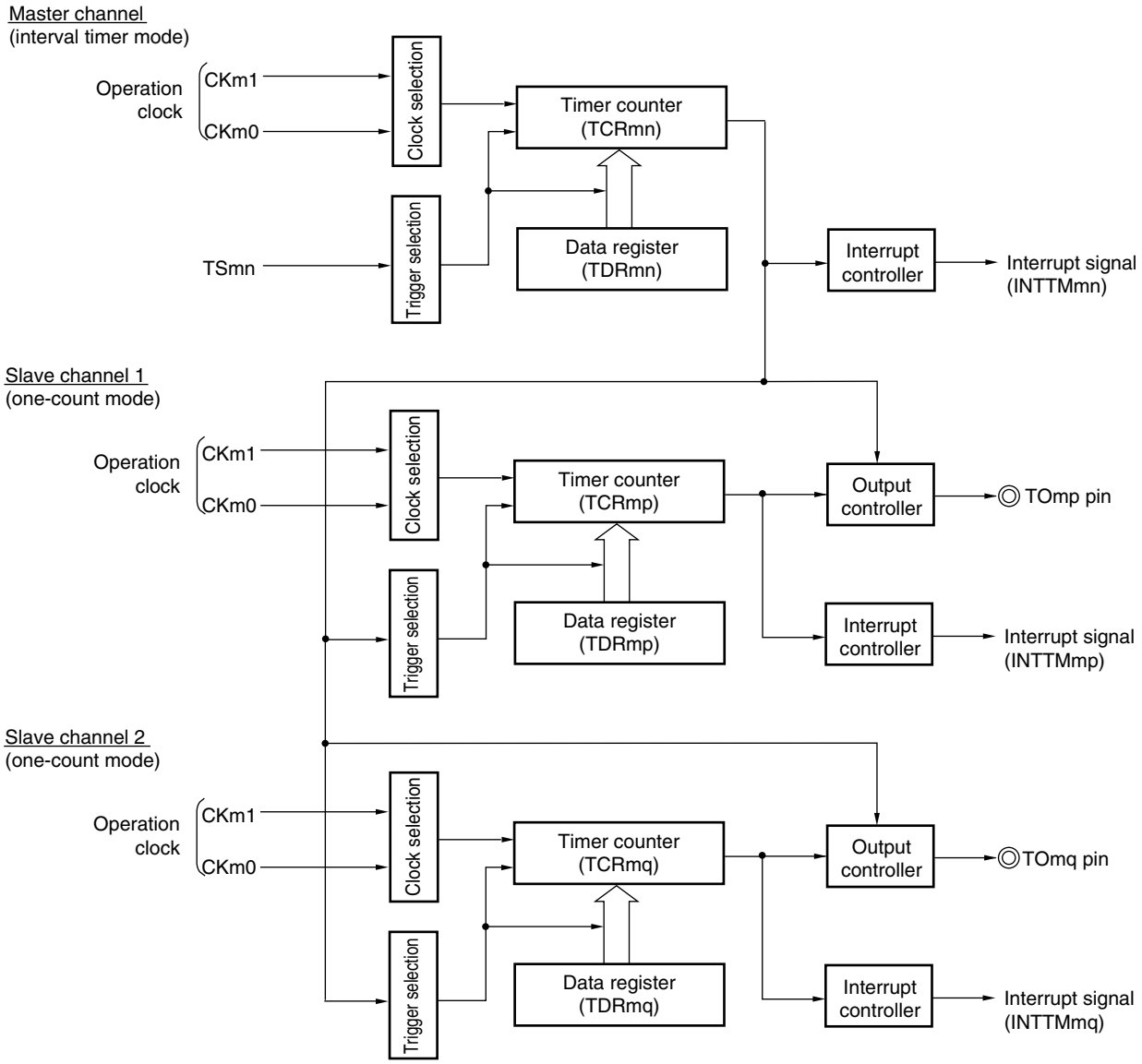
In the same way as TCRmp of the slave channel 1, TCRmq of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. TCRmq loads the value of TDRmq to TCRmq, using INTTMmn of the master channel as a start trigger, and starts counting down. When 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 two 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 ≤ 2 (where p and q are a consecutive integer greater than n (p = 1, q = 2))

Figure 6-67. 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

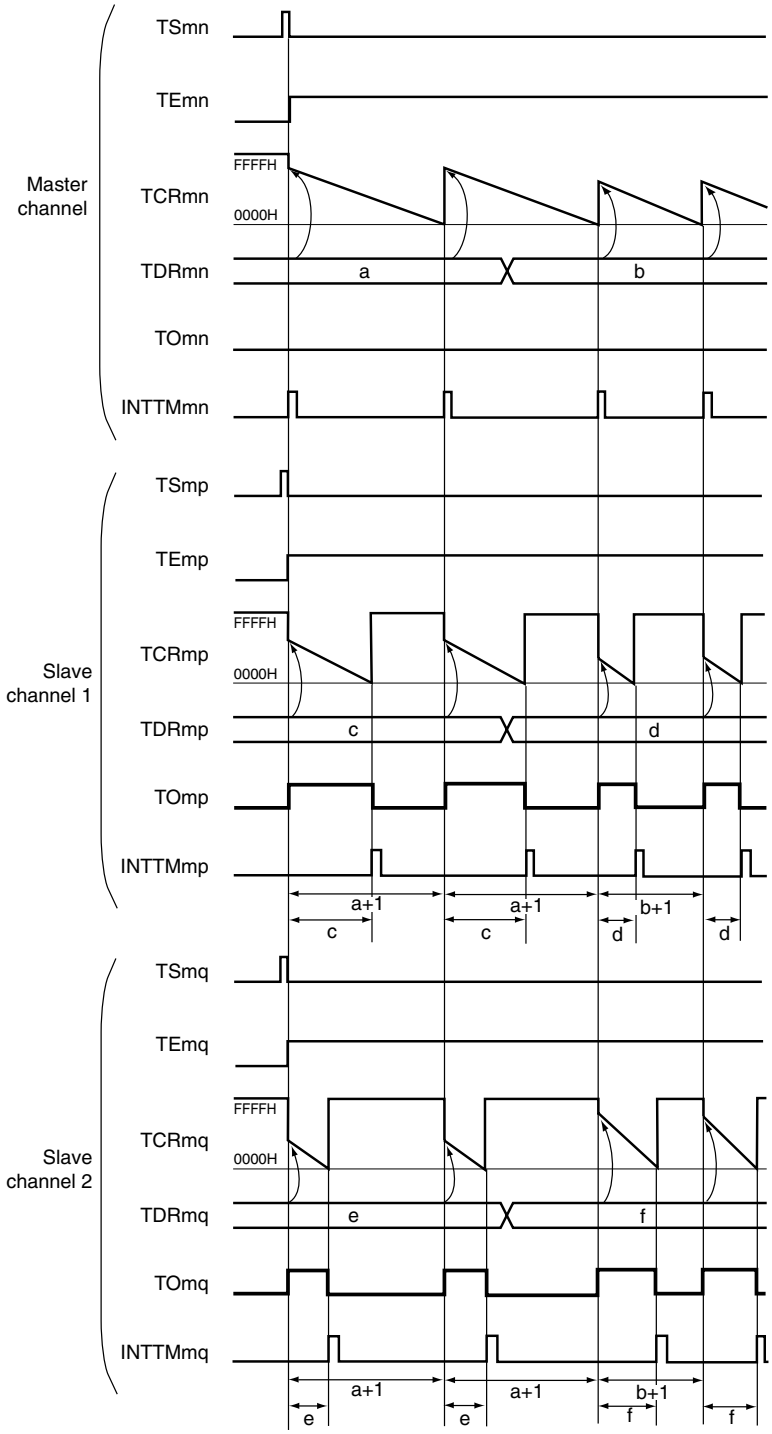
$n < p < q \leq 7$ (where p and q are a consecutive integer greater than n)

When m = 1

n = 0

$n < p < q \leq 2$ (where p and q are a consecutive integer greater than n (p = 1, q = 2))

Figure 6-68. 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)

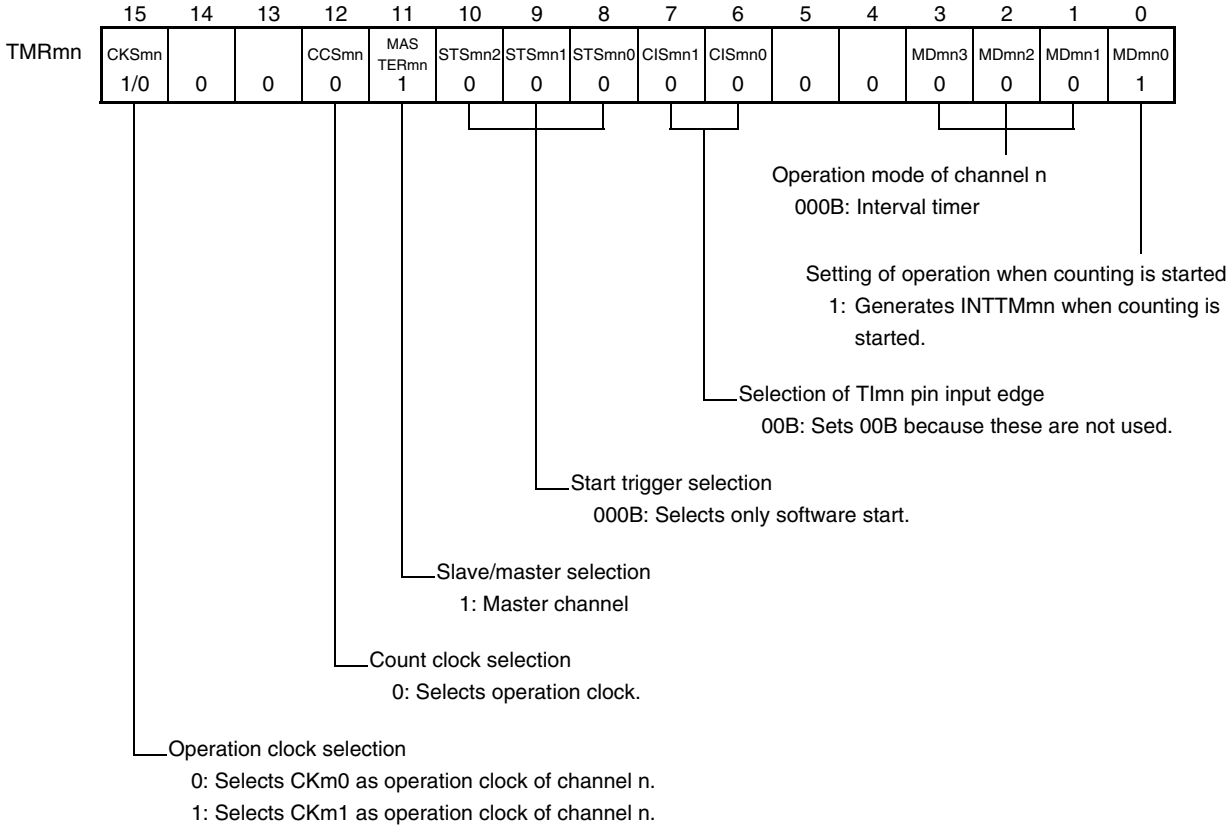
When m = 1

n = 0

n < p < q ≤ 2 (where p and q are a consecutive integer greater than n (p = 1, q = 2))

Figure 6-69. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)



(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Remark m: Unit number (m = 0, 1), n: Channel number
When m = 0: n = 0, 2, 4
When m = 1: n = 0

Figure 6-70. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)

(a) Timer mode register mp, mq (TMRmp, TMRmq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp			CCSmp	MAS TERmp	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1
TMRmq	CKSmq			CCSmq	MAS TERmq	STSmq2	STSmq1	STSmq0	CISmq1	CISmq0			MDmq3	MDmq2	MDmq1	MDmq0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p, q
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp and TImq pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Slave/master selection
0: Slave channel

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CKm0 as operation clock of channel p, q.
1: Selects CKm1 as operation clock of channel p, q.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit q	Bit p
TOm	TOmq	TOmp
	1/0	1/0

0: Outputs 0 from TOmp or TOmq.
1: Outputs 1 from TOmp or TOmq.

(c) Timer output enable register m (TOEm)

	Bit q	Bit p
TOEm	TOEmq	TOEmp
	1/0	1/0

0: Stops the TOmp or TOmq output operation by counting operation.
1: Enables the TOmp or TOmq output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit q	Bit p
TOLm	TOLmq	TOLmp
	1/0	1/0

0: Positive logic output (active-high)
1: Inverted output (active-low)

(e) Timer output mode register m (TOMm)

	Bit q	Bit p
TOMm	TOMmq	TOMmp
	1	1

1: Sets slave channel output mode.

(Remark is 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 ≤ 2 (where p and q are a consecutive integer greater than n (p = 1, q = 2))

Figure 6-71. 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 TOmq bits of the TOMm register is set to 1 (slave channel output mode). Clears the TOLmp and TOLmq bits to 0. Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. →	The TOmn pin goes into Hi-Z output state. The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets TOEmp and TOEmq to 1 and enables operation of TOmp and TOmq. →	TOmp or TOmq does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	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, TEmp, 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).

Figure 6-71. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

Operation is resumed (on the before page).

	Software Operation	Hardware Status
During operation	Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed. Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated. The TCRmn, TCRmp, and TCRmq registers can always be read. The TSRmn, TSRmp, and TSRmq registers are not used. Set values of the TOm and TOEm registers can be changed.	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 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. 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. After that, the above operation is repeated.
Operation stop	The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	TEmn, TEmp and TEmq = 0, and count operation stops. TCRmn, TCRmp and TCRmq hold count value and stops. The TOmp and TOmq output is not initialized but holds current status.
	TOEmp or TOEmq of slave channel is cleared to 0 and value is set to the TOmp and TOmq bits.	The TOmp and TOmq pins output the TOmp and TOmq set levels.
TAU stop	To hold the TOmp and TOmq pin output levels Clears TOmp and TOmq bits to 0 after the value to be held is set to the port register.	The TOmp and TOmq pin output levels are held by port function.
	When holding the TOmp and TOmq pin output levels is not necessary Switches the port mode register to input mode.	The TOmp and TOmq pin output levels go 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 and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.)

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 ≤ 2 (where p and q are a consecutive integer greater than n (p = 1, q = 2))

CHAPTER 7 REAL-TIME COUNTER

7.1 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

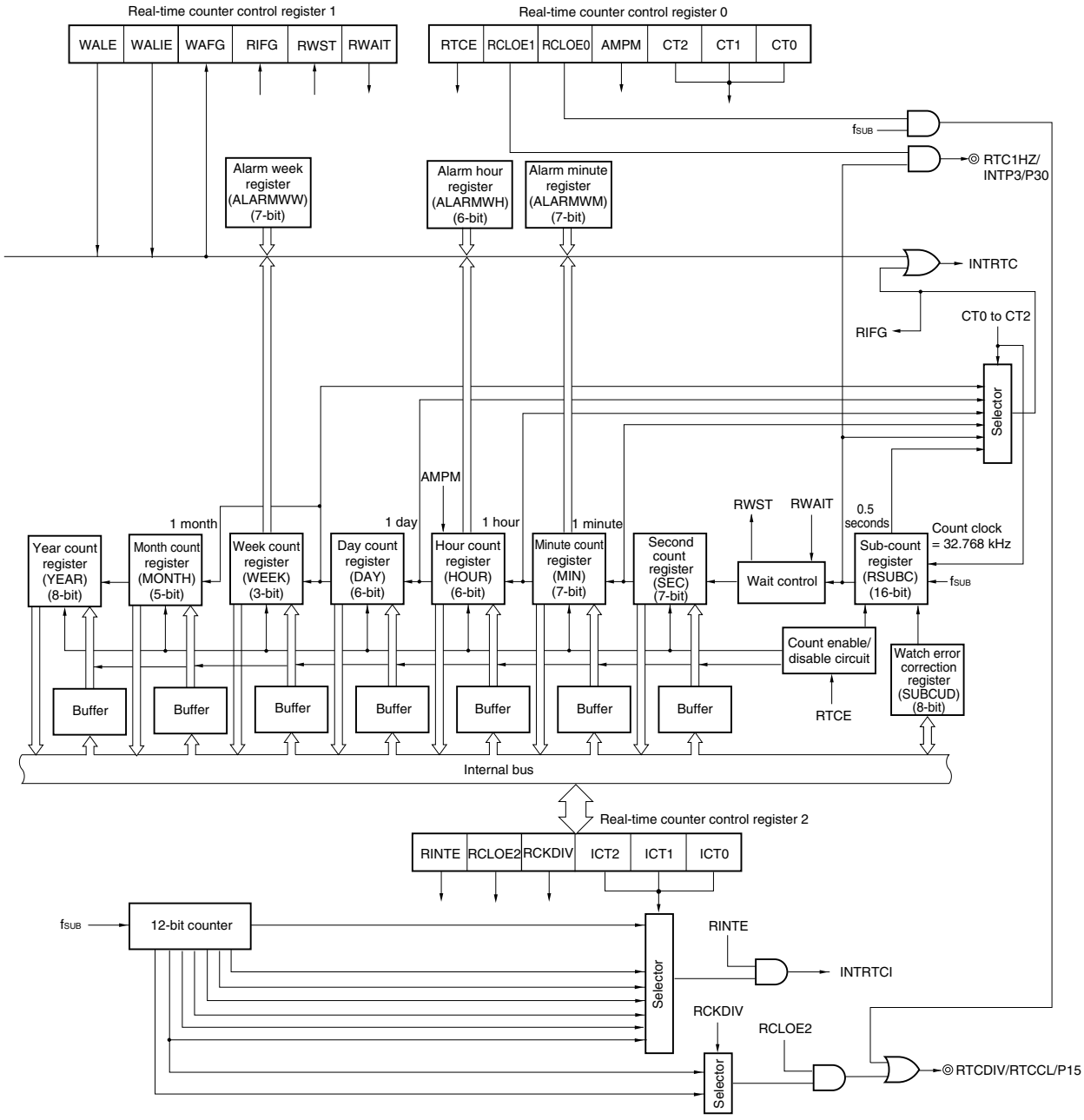
7.2 Configuration of Real-Time Counter

The real-time counter includes the following hardware.

Table 7-1. 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)

Figure 7-1. Block Diagram of Real-Time Counter



7.3 Registers Controlling Real-Time Counter

The real-time counter is controlled by the following 16 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, 3 (PM1, PM3)
- 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 7-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of real-time counter (RTC) input clock supply[Note]
0	Stops supply of input clock. • SFR used by the real-time counter (RTC) cannot be written. • The real-time counter (RTC) is in the reset status.
1	Supplies input clock. • SFR used by the real-time counter (RTC) can be read/written.

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.

Cautions 1. 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. Clock supply to peripheral functions other than the real-time counter can be stopped in HALT mode when the subsystem clock is used, by setting RTCLPC of the operation speed mode control register (OSMC) to 1. In that case, set RTCEN to 1 and bits 0 to 6 of PER0 to 0. Furthermore, set bits 0 to 7 of the PER1 register also to 0.

3. Be sure to clear bit 6 of the PER0 register to 0.

(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 7-3. 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[Note]	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
• To change the value of AMPM, set RWAIT (bit 0 of RTCC1) to 1. • When the AMPM value is changed, the value of the hour count register (HOUR) will be automatically changed to a value corresponding to the set time system. • Table 7-2 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, the last waveform of the 32.768 kHz and 1 Hz output signals may become short.

Remark ×: don't care

(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 7-4. 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 7-4. 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, 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 7-5. 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[Note]	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 7-6. 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 7-7. 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.

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

Reset signal generation clears this register to 00H.

Figure 7-8. 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 and 21 to 32 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 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.

Set a value of 00 to 23, or 01 to 12 and 21 to 32, in decimal format by using a BCD code, according to the time system set by using bit 3 (AMPM) of real-time counter control register 0 (RTCC0). If the value of the AMPM bit is changed, the HOUR value will be automatically changed to the corresponding value. 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 (bit 3 of the RTCC0 register) is set to 1 after reset.

Figure 7-9. 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 7-2 shows the relationship between the setting value of the AMPM bit, the HOUR register value, and time.

Table 7-2. 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

The HOUR register value is set to 12-hour display when the AMPM bit is “0” and to 24-hour display when the AMPM bit is “1”.

In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

(9) 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 7-10. 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 7-11. 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 7-12. 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 7-13. 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 7-14. 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. • When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H • When DEV = 1 is set: For a period of SEC = 00H	

F6	Setting of watch error correction value
0	Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.
When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected. * is 0 or 1. /F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100). Range of correction value: (when F6 = 0) 2, 4, 6, 8, … , 120, 122, 124 (when F6 = 1) –2, –4, –6, –8, … , –120, –122, –124	

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	–189.2 ppm to 189.2 ppm	–63.1 ppm to 63.1 ppm
Maximum excludes quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

Remark Set DEV to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(14) Alarm minute register (ALARMWM)

This register is used to set minutes of alarm.

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

Reset signal generation clears this register to 00H.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 7-15. Format of Alarm Minute Register (ALARMWM)

Address: FFF9AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

(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 7-16. 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 7-17. 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 WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

(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/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 7-18. 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

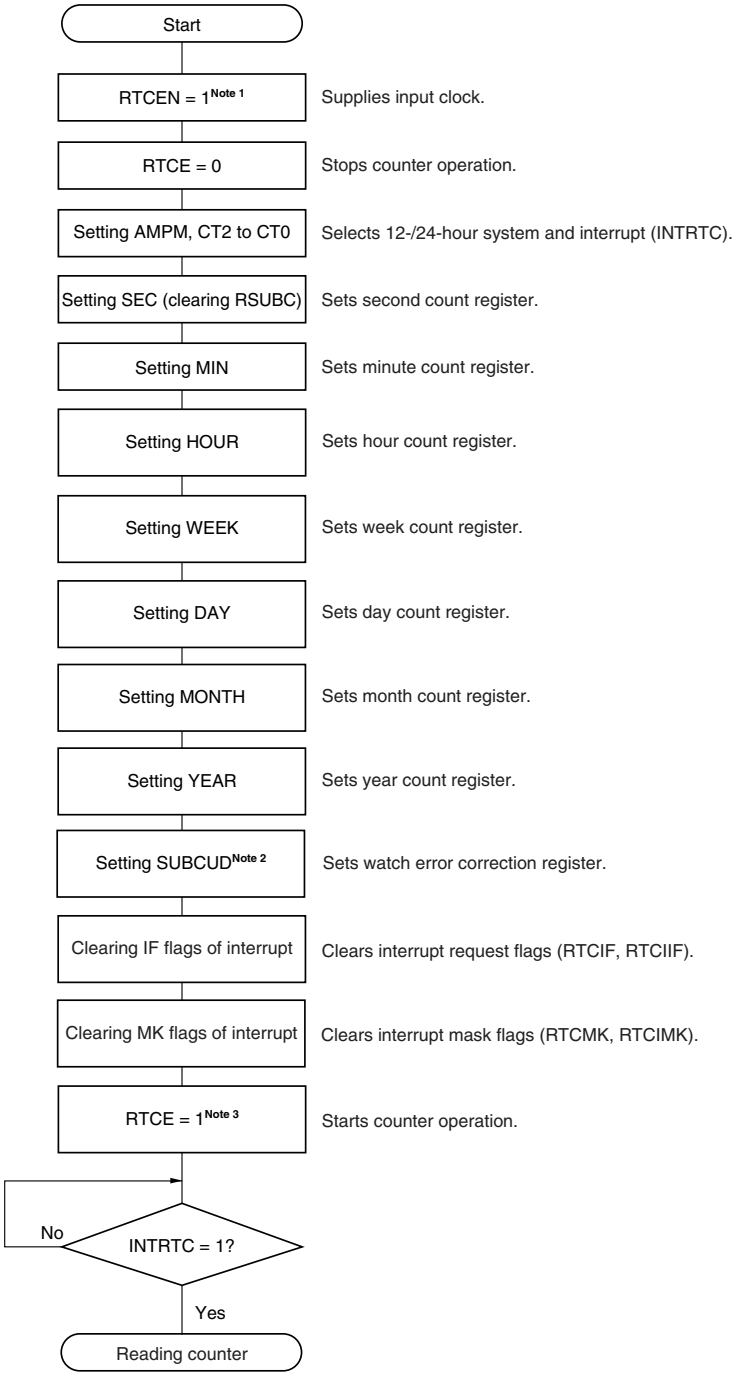
Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

PMmn	Pmn pin I/O mode selection (m = 1, 3 ; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.4 Real-Time Counter Operation

7.4.1 Starting operation of real-time counter

Figure 7-19. Procedure for Starting Operation of Real-Time Counter



Notes 1. First set RTCEN to 1, while oscillation of the subsystem clock (f_{SUB}) is stable.

2. Set up the SUBCUD register only if the watch error must be corrected. For details about how to calculate the correction value, see **7.4.8 Example of watch error correction of real-time counter**.

3. Confirm the procedure described in **7.4.2 Shifting to STOP mode after starting operation** when shifting to STOP mode without waiting for INTRTC = 1 after RTCE = 1.

7.4.2 Shifting to STOP mode after starting operation

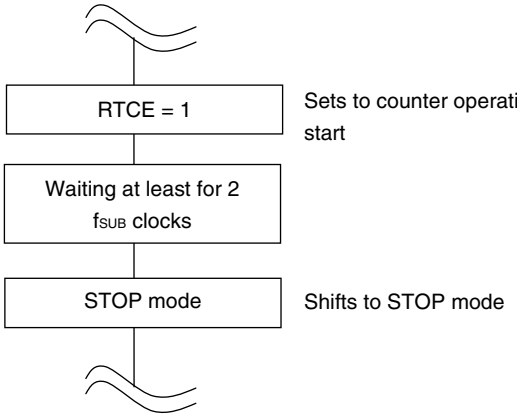
Perform one of the following processing when shifting to STOP mode immediately after setting RTCE to 1.

However, after setting RTCE to 1, this processing is not required when shifting to STOP mode after the first INTRTC interrupt has occurred.

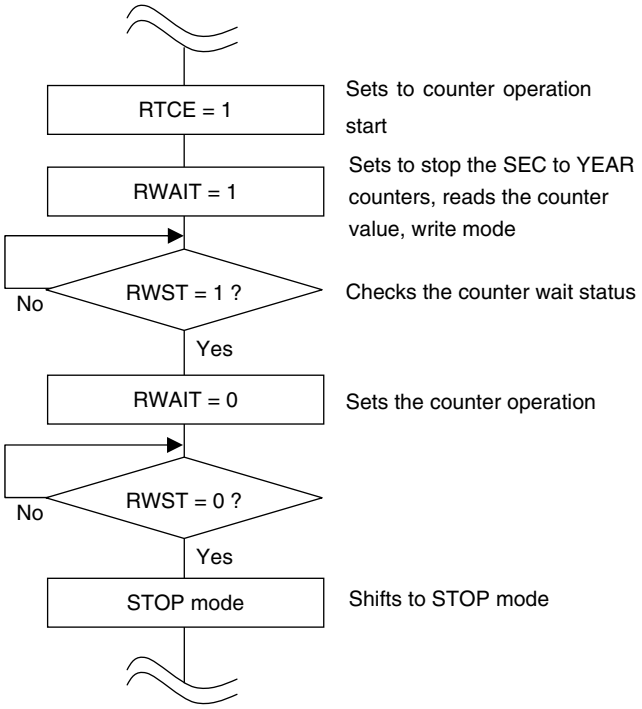
- Shifting to STOP mode when at least two subsystem clocks (f_{SUB}) (about 62 μs) have elapsed after setting RTCE to 1 (see **Figure 7-20**, **Example 1**).
- Checking by polling RWST to become 1, after setting RTCE to 1 and then setting RWAIT to 1. Afterward, setting RWAIT to 0 and shifting to STOP mode after checking again by polling that RWST has become 0 (see **Figure 7-20**, **Example 2**).

Figure 7-20. Procedure for Shifting to STOP Mode After Setting RTCE to 1

Example 1



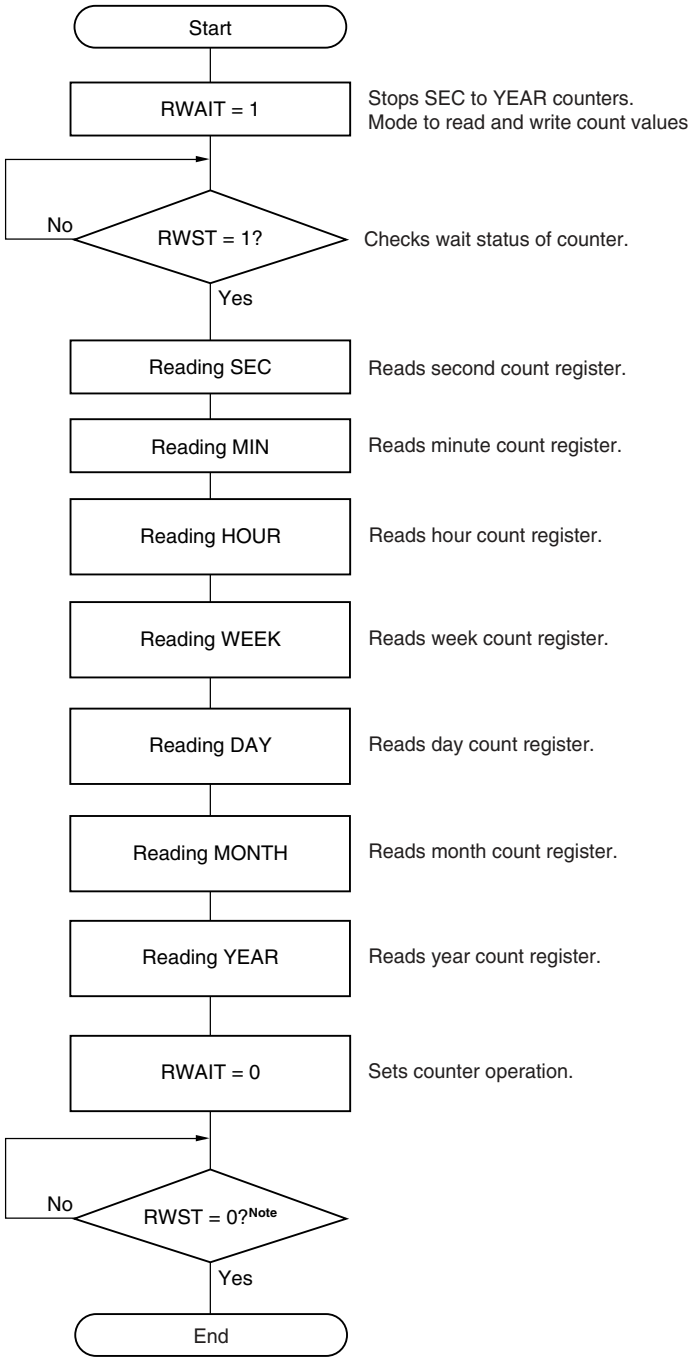
Example 2



7.4.3 Reading/writing real-time counter

Read or write the counter after setting 1 to RWAIT first.

Figure 7-21. Procedure for Reading Real-Time Counter

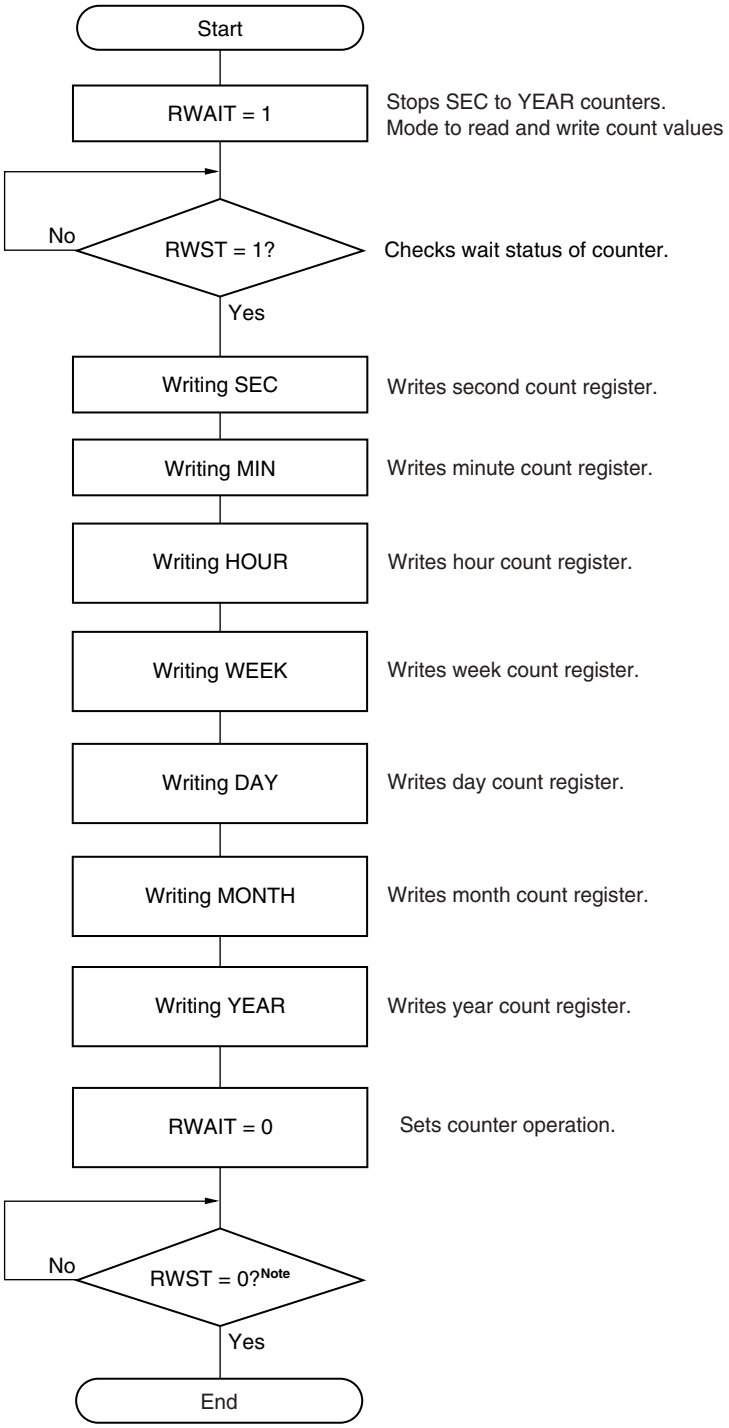


Note Be sure to confirm that RWST = 0 before setting STOP mode.

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 7-22. Procedure for Writing Real-Time Counter



Note Be sure to confirm that RWST = 0 before setting STOP mode.

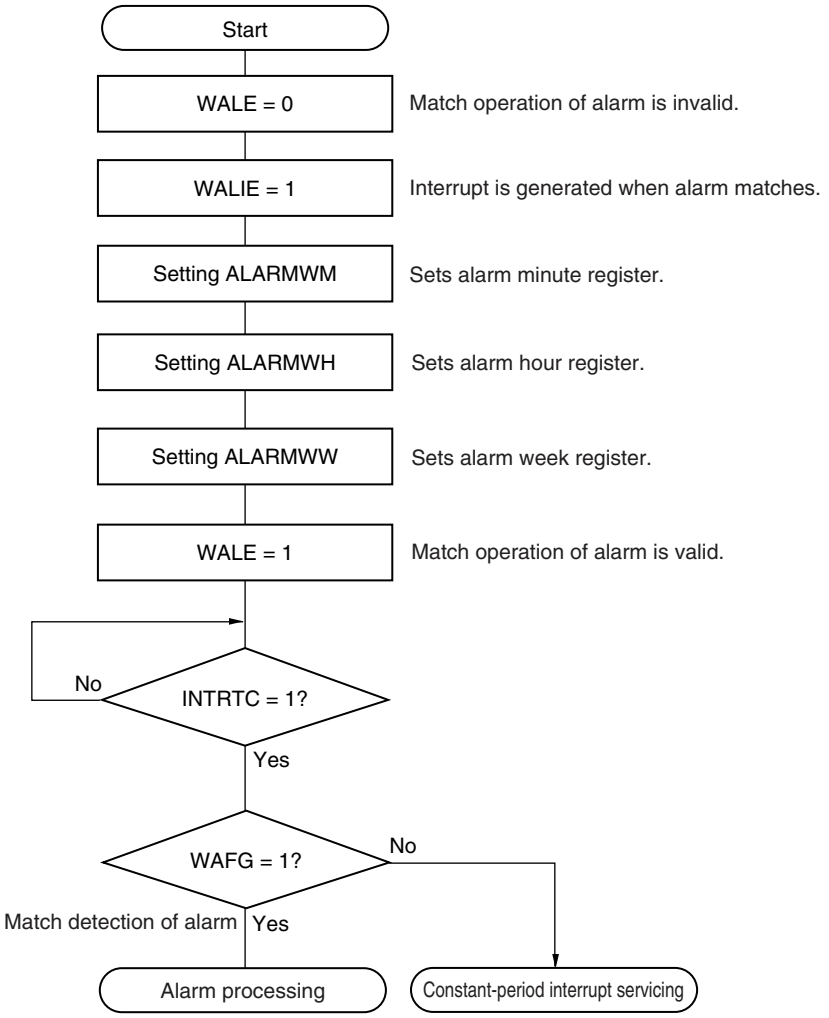
Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

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.

7.4.4 Setting alarm of real-time counter

Set time of alarm after setting 0 to WALE first.

Figure 7-23. Alarm Setting Procedure

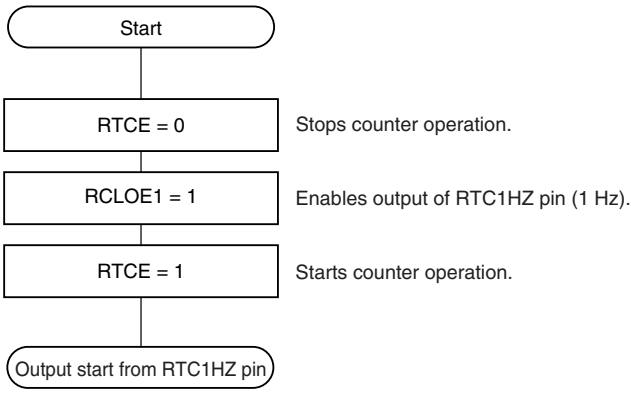


Remarks 1. ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

7.4.5 1 Hz output of real-time counter

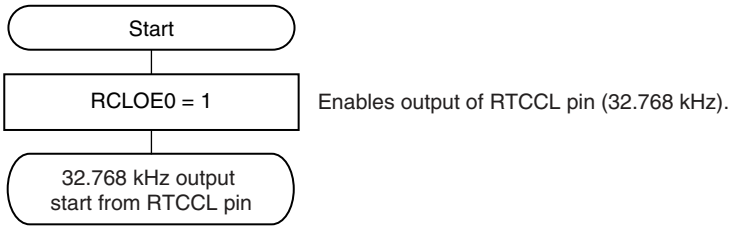
Figure 7-24. 1 Hz Output Setting Procedure



Caution First set the RTCEN bit to 1, while oscillation of the subsystem clock (f_{SUB}) is stable.

7.4.6 32.768 kHz output of real-time counter

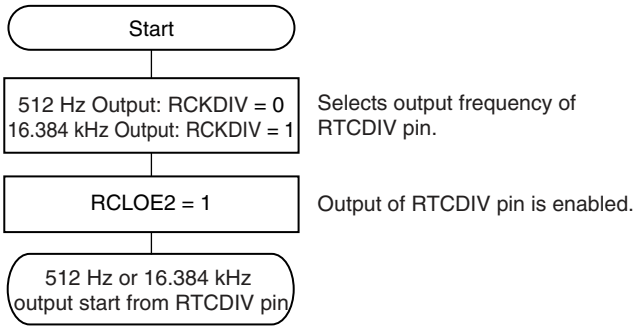
Figure 7-25. 32.768 kHz Output Setting Procedure



Caution First set the RTCEN bit to 1, while oscillation of the subsystem clock (f_{SUB}) is stable.

7.4.7 512 Hz, 16.384 kHz output of real-time counter

Figure 7-26. 512 Hz, 16.384 kHz output Setting Procedure



Caution First set the RTCEN bit to 1, while oscillation of the subsystem clock (f_{SUB}) is stable.

7.4.8 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)

Correction value[Note] = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60 ÷ 3

(When DEV = 1)

Correction value[Note] = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60

Note The correction value is the watch error correction value calculated by using bits 6 to 0 of the watch error correction register (SUBCUD).

(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) – 1} × 2

(When F6 = 1) Correction value = – {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), watch error correction is not performed. "*" is 0 or 1. /F5 to /F0 are bit-inverted values (000011 when 111100).

Remarks 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

2. The oscillation frequency is the subsystem clock (f_{SUB}).
It can be calculated from the 32 kHz output frequency of the RTCCL pin or the output frequency of the RTC1HZ pin × 32768 when the watch error correction register is set to its initial value (00H).

3. The target frequency is the frequency resulting after correction performed by using the watch error correction register.

Correction example <1>

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32 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 **7.4.5 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **7.4.6 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.

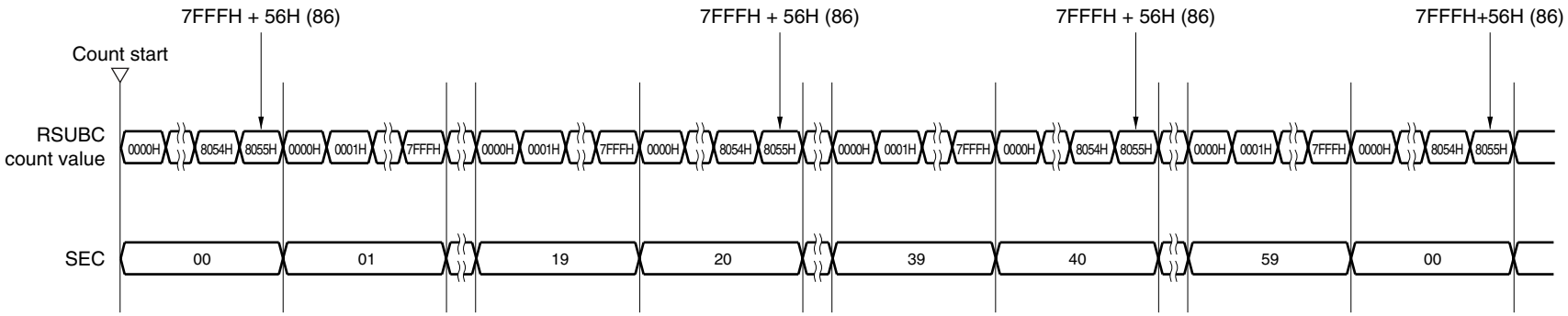
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

$$\begin{aligned}\{(F5, F4, F3, F2, F1, F0) - 1\} \times 2 &= 86\\ (F5, F4, F3, F2, F1, F0) &= 44\\ (F5, F4, F3, F2, F1, F0) &= (1, 0, 1, 1, 0, 0)\end{aligned}$$

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 7-27 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 7-27. 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[Note] 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 **7.4.5 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **7.4.6 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 × 0.9999817 ≈ 32767.4 Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

Correction value = Number of correction counts in 1 minute
= (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60
= (32767.4 ÷ 32768 − 1) × 32768 × 60
= −36

[Calculating the values to be set to (F6 to F0)]

(When the correction value is −36)

If the correction value is 0 or less (when speeding up), assume F6 to be 1.

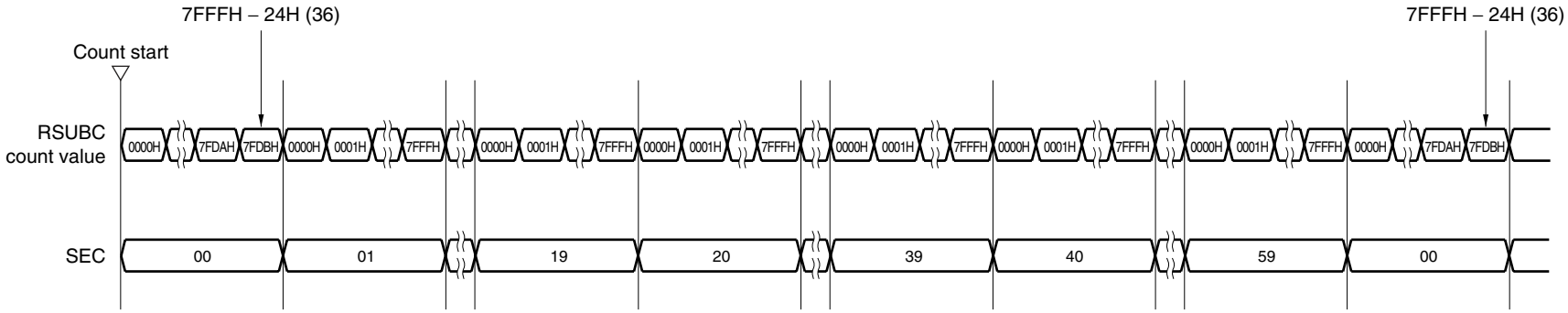
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

− {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2	= −36
(/F5, /F4, /F3, /F2, /F1, /F0)	= 17
(/F5, /F4, /F3, /F2, /F1, /F0)	= (0, 1, 0, 0, 0, 1)
(F5, F4, F3, F2, F1, F0)	= (1, 0, 1, 1, 1, 0)

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is −36 (bits 6 to 0 of SUBCUD: 1101110) results in 32768 Hz (0 ppm).

Figure 7-28 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

Figure 7-28. Operation When (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)



CHAPTER 8 WATCHDOG TIMER

8.1 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.

8.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 8-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

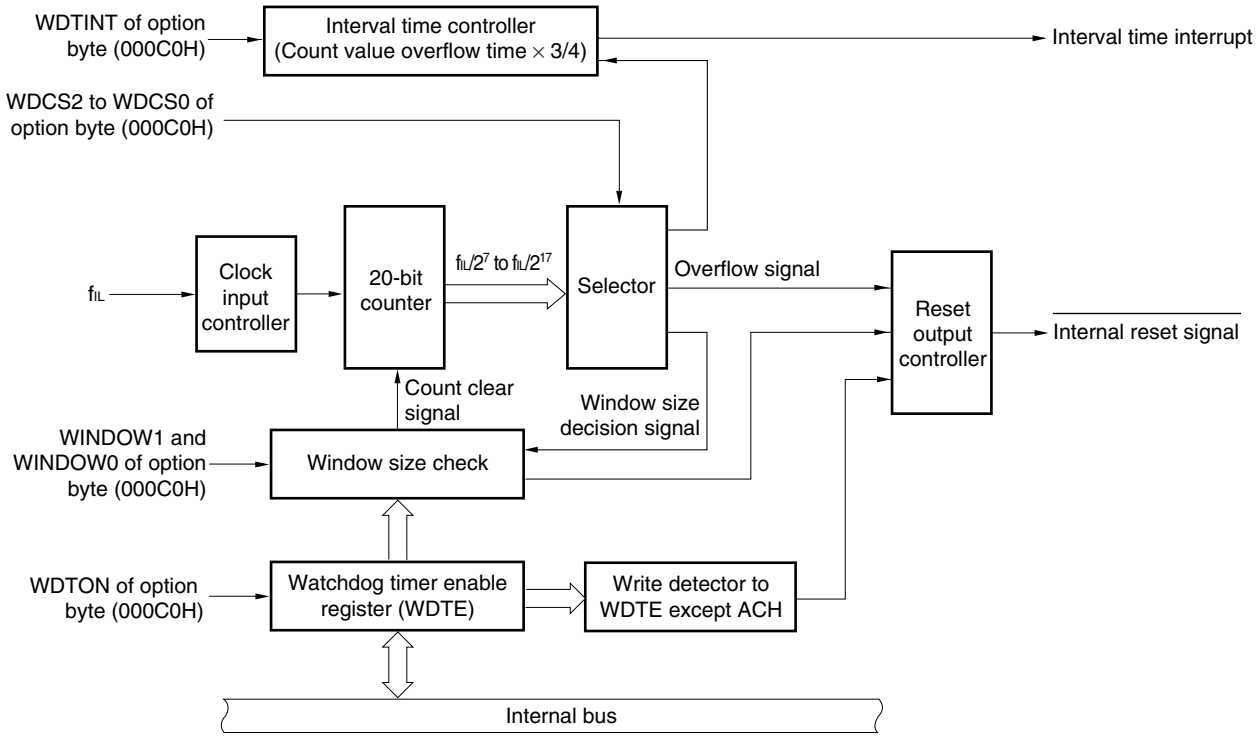
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 8-2. 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 8-1. Block Diagram of Watchdog Timer



8.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

(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[Note].

Figure 8-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH After reset: 9AH/1AH[Note] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE 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)).

8.4 Operation of Watchdog Timer

8.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 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 **8.4.2** 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 **8.4.3** 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.
   An 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.

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.

8.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to WDTE during the window open period before the overflow time.

The following overflow times can be set.

Table 8-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 33 kHz (MAX.))
0	0	0	$2^7/f_{IL}$ (3.88 ms)
0	0	1	$2^8/f_{IL}$ (7.76 ms)
0	1	0	$2^9/f_{IL}$ (15.52 ms)
0	1	1	$2^{10}/f_{IL}$ (31.03 ms)
1	0	0	$2^{12}/f_{IL}$ (124.12 ms)
1	0	1	$2^{14}/f_{IL}$ (496.48 ms)
1	1	0	$2^{15}/f_{IL}$ (992.97 ms)
1	1	1	$2^{17}/f_{IL}$ (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.

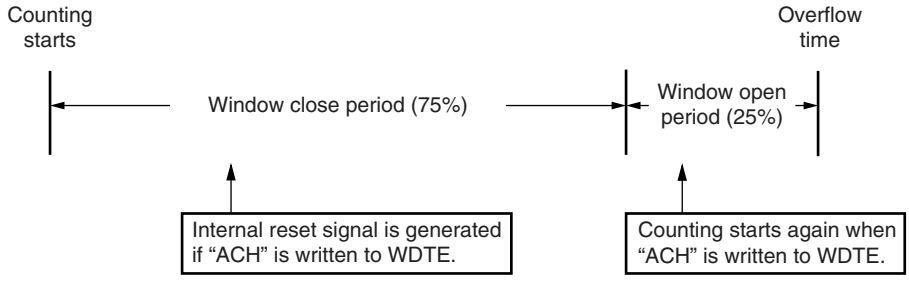
Remark f_{IL}: Internal low-speed oscillation clock frequency

8.4.3 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 can be set is as follows.

Table 8-4. 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 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.**

Remark 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 28.44 ms	0 to 18.96 ms	0 to 9.48 ms	None
Window open time	28.44 to 31.03 ms	18.96 to 31.03 ms	9.48 to 31.03 ms	0 to 31.03 ms

<When window open period is 25%>

- Overflow time:
  $2^{10}/f_{IL}$ (MAX.) = 2^{10}/33 kHz (MAX.) = 31.03 ms
- Window close time:
  0 to $2^{10}/f_{IL}$ (MIN.) × (1 − 0.25) = 0 to 2^{10}/27 kHz (MIN.) × 0.75 = 0 to 28.44 ms
- Window open time:
  $2^{10}/f_{IL}$ (MIN.) × (1 − 0.25) to $2^{10}/f_{IL}$ (MAX.) = 2^{10}/27 kHz (MIN.) × 0.75 to 2^{10}/33 kHz (MAX.) = 28.44 to 31.03 ms

8.4.4 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 8-5. 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 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

9.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral ICs.

Buzzer output is a function to output a square wave of buzzer frequency.

One pin can be used to output a clock or buzzer sound.

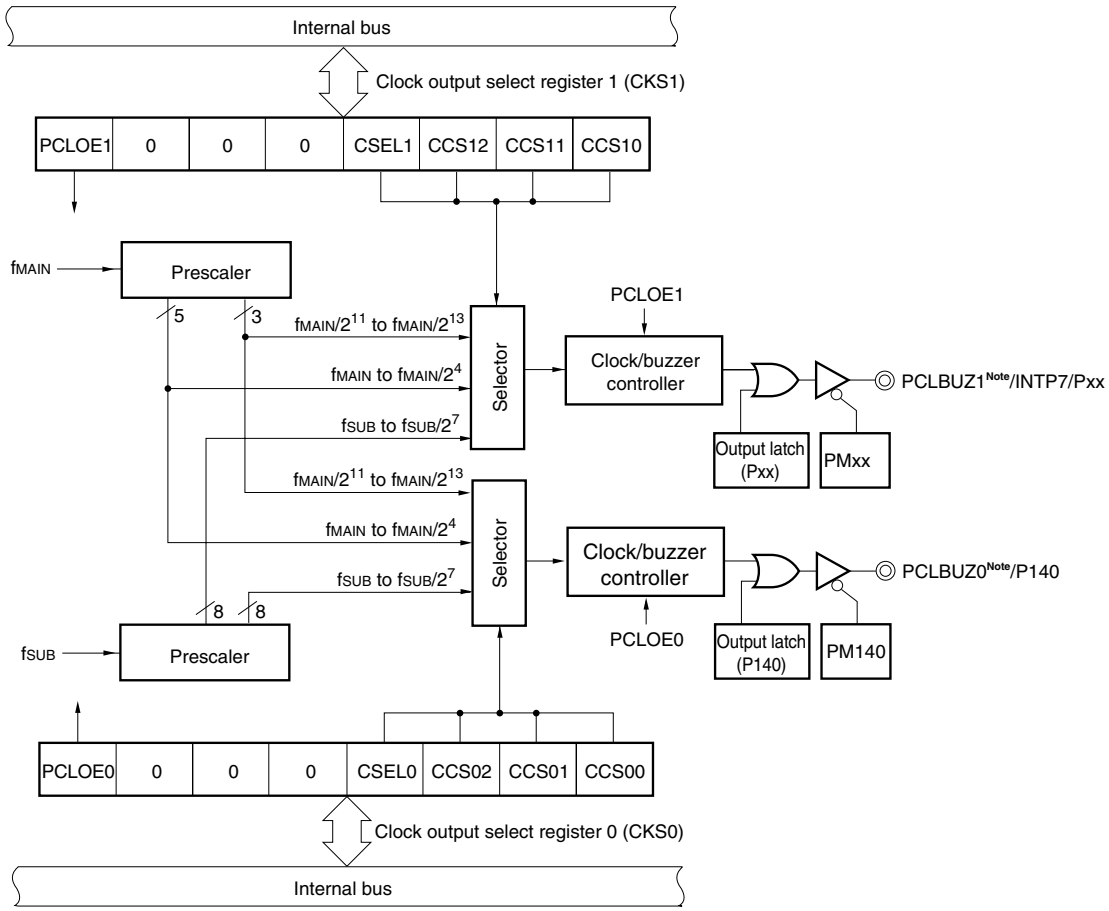
Two output pins, PCLBUZ0 and PCLBUZ1, are available.

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 9-1 shows the block diagram of clock output/buzzer output controller.

Figure 9-1. 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 V ≤ V_{DD}.

Remarks 1. xx = 55 (78K0R/KF3-C), xx = 141 (78K0R/KG3-C)

2. f_{MAIN}: Main system clock frequency

f_{SUB}: Subsystem clock frequency

9.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 9-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	• 78K0R/KF3-C Clock output select registers 0, 1 (CKS0, CKS1) Port mode registers 5, 14 (PM5, PM14) Port registers 5, 14 (P5, P14) • 78K0R/KG3-C Clock output select registers 0, 1 (CKS0, CKS1) Port mode register 14 (PM14) Port register 14 (P14)

9.3 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 (PMxx)

(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 9-2. Format of Clock Output Select Register n (CKSn)

Address: FFFA5H (CKS0), FFFA6H (CKS1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn output enable/disable specification
0	Output disable (default)
1	Output enable

CSELn	CCSn2	CCSn1	CCSn0		PCLBUZn output clock selection		
					f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz
0	0	0	0	f_{MAIN}	5 MHz	10 MHz	Setting prohibited[Note]
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.76 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz
1	0	0	0	f_{SUB}	32.768 kHz		
1	0	0	1	$f_{SUB}/2$	16.384 kHz		
1	0	1	0	$f_{SUB}/2^2$	8.192 kHz		
1	0	1	1	$f_{SUB}/2^3$	4.096 kHz		
1	1	0	0	$f_{SUB}/2^4$	2.048 kHz		
1	1	0	1	$f_{SUB}/2^5$	1.024 kHz		
1	1	1	0	$f_{SUB}/2^6$	512 Hz		
1	1	1	1	$f_{SUB}/2^7$	256 Hz		

Note Setting an output clock exceeding 10 MHz is prohibited.

Cautions 1. Change the output clock after disabling clock output (PCLOEn = 0).

2. To shift to STOP mode when the main system clock is selected (CSELn = 0), set PCLOEn = 0 before executing the STOP instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn = 1 can be set because the clock can be output in STOP mode.

Remarks 1. n = 0, 1

2. f_{MAIN}: Main system clock frequency

f_{SUB}: Subsystem clock frequency

(2) Port mode registers (PMxx)

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

When using the P140/PCLBUZ0 and Pxx/INTP7/PCLBUZ1 pins for clock output/buzzer output, clear PM140 and PM141 and the output latches of P140 and Pxx to 0.

PM14 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Remark xx = 55 (78K0R/KF3-C), xx = 141 (78K0R/KG3-C)

Figure 9-3. Format of Port Mode Register (78K0R/KF3-C)

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	1	PM144	PM143	PM142	1	PM140

PMmn	Pmn pin I/O mode selection (m = 5, 14; n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Figure 9-4. Format of Port Mode Register (78K0R/KG3-C)

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)

9.4 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).

9.4.1 Operation as output pin

PCLBUZn is output as the following procedure.

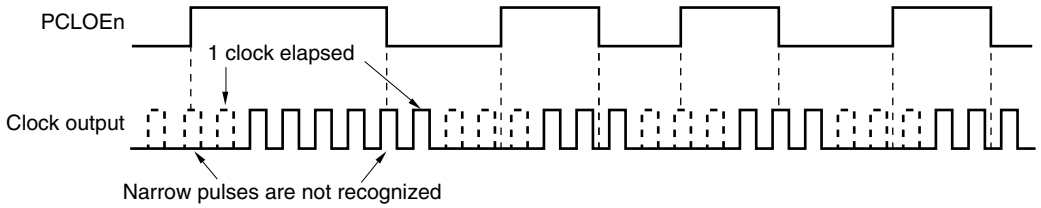
<1> Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of the clock output select register (CKSn) of the PCLBUZn pin (output in disabled status).

<2> Set bit 7 (PCLOEn) of CKSn to 1 to enable clock/buzzer output.

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 9-5 shows enabling or stopping output using PCLOEn and the timing of outputting the clock.

2. n = 0, 1

Figure 9-5. Remote Control Output Application Example



CHAPTER 10 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

	78K0R/KF3-C (μPD78F1846A, 78F1847A)	78K0R/KG3-C (μPD78F1848A, 78F1849A)
Analog input channels	12 ch (ANI0 to ANI11)	16 ch (ANI0 to ANI15)

10.1 Function of A/D Converter

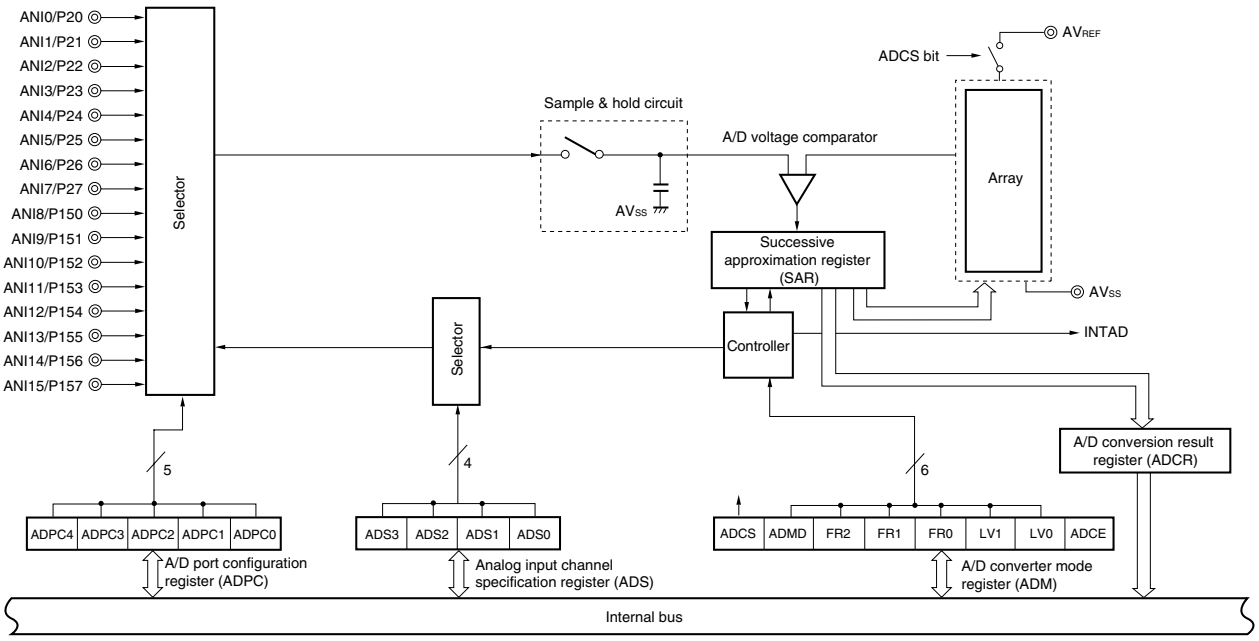
The A/D converter converts an analog input signal into a digital value, and consists of up to 16 channels (ANI0 to ANI15) with a resolution of 10 bits.

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 10-1. Block Diagram of A/D Converter



Remark ANI0 to ANI11: 78K0R/KF3-C
ANI0 to ANI15: 78K0R/KG3-C

10.2 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 channels of the A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

Remark ANI0 to ANI11: 78K0R/KF3-C
ANI0 to ANI15: 78K0R/KG3-C

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

(3) A/D voltage comparator

This A/D voltage comparator compares the voltage generated from the voltage tap of the array with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REF}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AV_{REF}), the MSB of the SAR is reset.
After that, bit 10 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the array is selected by the value of bit 11, to which the result has been already set.

Bit 11 = 0: (1/4 AV_{REF})
Bit 11 = 1: (3/4 AV_{REF})

The voltage tap of the array and the analog input voltage are compared and bit 10 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of array: Bit 10 = 1
Analog input voltage ≤ Voltage tap of array: Bit 10 = 0

Comparison is continued like this to bit 0 of the SAR register.

(4) Array

The array generates the comparison voltage input from an analog input pin.

(5) Successive approximation register (SAR)

The SAR register is a 12-bit register that sets voltage tap data whose values from the array match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).
If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

6) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD.

(9) AV_{REF} pin

This pin inputs the reference voltage of the A/D converter, the power supply pins and A/D converter of the comparator. When all pins of ports 2 and 15 are used as the analog port pins, make the potential of AV_{REF} be such that 2.7 V ≤ AV_{REF} ≤ V_{DD}. When one or more of the pins of ports 2 and 15 are used as the digital port pins, make AV_{REF} the same potential as V_{DD}.

The analog signal input to ANI0 to ANI10 is converted into a digital signal, based on the voltage applied across AV_{REF} and AV_{SS}.

Remark ANI0 to ANI11: 78K0R/KF3-C
ANI0 to ANI15: 78K0R/KG3-C

(10) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always use this pin at the same potential as that of the V_{SS} pin even when the A/D converter is not used.

10.3 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, 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 10-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

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

Cautions 1. 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.

2. Be sure to clear bit 6 of PER0 register to 0.

(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 10-3. 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	ADMD	FR2 Note 1	FR1 Note 1	FR0 Note 1	LV1 Note 1	LV0 Note 1	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation
1	Enables conversion operation

ADMD	A/D conversion operation mode specification
0	Select mode
1	Scan mode

ADCE	A/D voltage comparator operation control Note 2
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

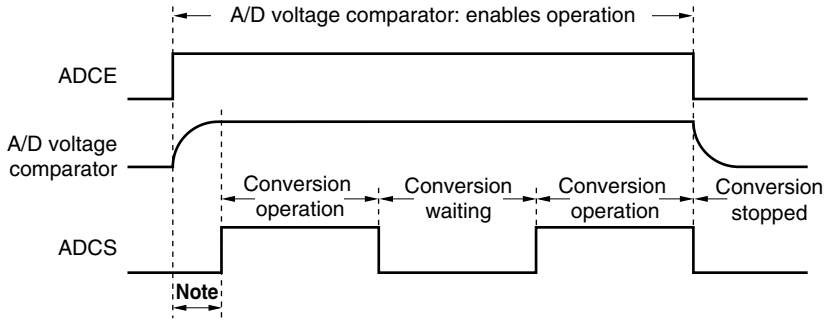
Notes 1. For details of FR2 to FR0, LV1, LV0, and A/D conversion, see **Table 10-2 A/D Conversion Time Selection**.

2. The operation of the A/D voltage comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, by waiting for at least 1 μs to elapse before setting ADCS to 1 after ADCE has been set to 1, the conversion results are valid from the first result. Otherwise, ignore data of the first conversion.

Table 10-1. 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 comparato: enables operation)

Figure 10-4. 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 rising of the ADCS bit must be 1 μs 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 10-2. A/D Conversion Time Selection (1/2)

(1) 4.0 V ≤ AV_{REF} ≤ 5.5 V

A/D Converter Mode Register (ADM)					Mode	Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	
0	0	0	0	0	Normal	Setting prohibited	Setting prohibited	34.2 μs	17.1 μs	f_{CLK}/20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f_{CLK}/10
0	1	0					27.6 μs	13.8 μs	6.9 μs	f_{CLK}/8
0	1	1				52.0 μs	20.8 μs	10.4 μs	5.2 μs	f_{CLK}/6
1	0	0				35.0 μs	14.0 μs	7.0 μs	Setting prohibited	f_{CLK}/4
1	0	1				26.5 μs	10.6 μs	5.3 μs		f_{CLK}/3
1	1	0				18.0 μs	7.2 μs	Setting prohibited		f_{CLK}/2
1	1	1				9.5 μs	Setting prohibited			f_{CLK}
×	×	×	0	1	Low-voltage	Setting prohibited				–
0	0	0	1	0	High speed 1	Setting prohibited	64.4 μs	32.2 μs	16.1 μs	f_{CLK}/20
0	0	1					32.4 μs	16.2 μs	8.1 μs	f_{CLK}/10
0	1	0				65.0 μs	26.0 μs	13.0 μs	6.5 μs	f_{CLK}/8
0	1	1				49.0 μs	19.6 μs	9.8 μs	4.9 μs	f_{CLK}/6
1	0	0				33.0 μs	13.2 μs	6.6 μs	3.3 μs	f_{CLK}/4
1	0	1				25.0 μs	10.0 μs	5.0 μs	2.5 μs	f_{CLK}/3
1	1	0				17.0 μs	6.8 μs	3.4 μs	Setting prohibited	f_{CLK}/2
1	1	1				9.0 μs	3.6 μs	Setting prohibited		f_{CLK}
0	0	0	1	1	High speed 2	Setting prohibited	Setting prohibited	34.2 μs	17.1 μs	f_{CLK}/20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f_{CLK}/10
0	1	0					27.6 μs	13.8 μs	6.9 μs	f_{CLK}/8
0	1	1				52.0 μs	20.8 μs	10.4 μs	5.2 μs	f_{CLK}/6
1	0	0				35.0 μs	14.0 μs	7.0 μs	3.5 μs	f_{CLK}/4
1	0	1				26.5 μs	10.6 μs	5.3 μs	Setting prohibited	f_{CLK}/3
1	1	0				18.0 μs	7.2 μs	3.6 μs		f_{CLK}/2
1	1	1				9.5 μs	3.8 μs	Setting prohibited		f_{CLK}

Cautions 1. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. 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

Table 10-2. A/D Conversion Time Selection (2/2)

(2) 2.7 V ≤ AV_{REF} ≤ 5.5 V

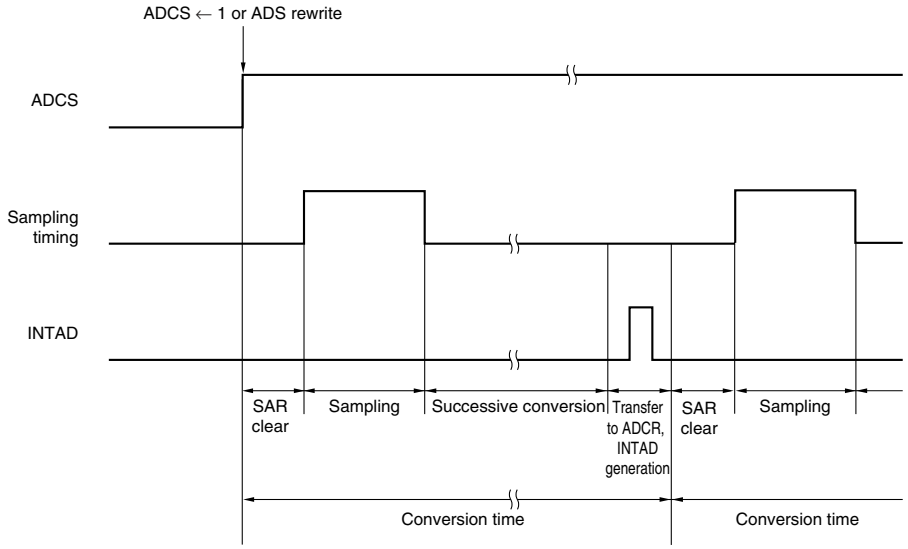
A/D Converter Mode Register (ADM)					Mode	Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	
0	0	0	0	0	Normal	Setting prohibited	Setting prohibited	34.2 μs	17.1 μs	f_{CLK}/20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f_{CLK}/10
0	1	0					27.6 μs	13.8 μs	Setting prohibited	f_{CLK}/8
0	1	1				52.0 μs	20.8 μs	10.4 μs		f_{CLK}/6
1	0	0				35.0 μs	14.0 μs	Setting prohibited		f_{CLK}/4
1	0	1				26.5 μs	10.6 μs			f_{CLK}/3
1	1	0				18.0 μs	Setting prohibited			f_{CLK}/2
1	1	1				9.5 μs				f_{CLK}
×	×	×	0	1	Low-voltage	Setting prohibited				–
×	×	×	1	0	High speed 1	Setting prohibited				–
0	0	0	1	1	High speed 2	Setting prohibited	Setting prohibited	34.2 μs	17.1 μs	f_{CLK}/20
0	0	1					34.4 μs	17.2 μs	8.6 μs	f_{CLK}/10
0	1	0					27.6 μs	13.8 μs	6.9 μs	f_{CLK}/8
0	1	1				52.0 μs	20.8 μs	10.4 μs	5.2 μs	f_{CLK}/6
1	0	0				35.0 μs	14.0 μs	7.0 μs	3.5 μs	f_{CLK}/4
1	0	1				26.5 μs	10.6 μs	5.3 μs	Setting prohibited	f_{CLK}/3
1	1	0				18.0 μs	7.2 μs	3.6 μs		f_{CLK}/2
1	1	1				9.5 μs	3.8 μs	Setting prohibited		f_{CLK}

Cautions 1. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

2. 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 10-5. 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 in the select mode. 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 10-6. Format of 10-bit A/D Conversion Result Register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR											0	0	0	0	0	0

Caution When writing to 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 10-7. Format of 8-bit A/D Conversion Result Register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to 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 10-8. Format of Analog Input Channel Specification Register (ADS) (1/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0)

KG3-C	KF3-C	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
Note 1	Note 1	0	0	0	0	ANI0	P20/ANI0 pin
		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
	Note 2	1	1	0	1	ANI13	P155/ANI13 pin
		1	1	1	0	ANI14	P156/ANI14 pin
		1	1	1	1	ANI15	P157/ANI15 pin
		Other than the above				Setting prohibited	

Notes 1. Setting permitted

2. Setting prohibited

Cautions 1. Be sure to clear bits 4 to 7 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 and PM15).

3. Do not set the pin that is set by ADPC as digital I/O by ADS.

Remark P20/ANI0 to P27/ANI7, P150/ANI8 to P153/ANI11: 78K0R/KF3-C

P20/ANI0 to P27/ANI7, P150/ANI8 to P157/ANI15: 78K0R/KG3-C

Figure 10-8. Format of Analog Input Channel Specification Register (ADS) (2/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

○ Scan mode (ADMD = 1)

ADS3	ADS2	ADS1	ADS0	Analog input channel			
				Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	1	0	1	ANI5	ANI6	ANI7	ANI8
0	1	1	0	ANI6	ANI7	ANI8	ANI9
0	1	1	1	ANI7	ANI8	ANI9	ANI10
Other than the above				Setting prohibited			

Cautions 1. Be sure to clear bits 4 to 7 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.

(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 10-9. 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	A
0	0	0	0	1	A	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	A	D	D
0	0	0	1	1	A	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	A	D	D	D	D
0	0	1	0	1	A	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	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	D
0	1	0	0	0	A	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	D
0	1	0	1	0	A	A	A	A	A	A	D	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	D
0	1	1	0	0	A	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	D
0	1	1	1	0	A	A	D	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	D
1	0	0	0	0	D	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 register 2 and 15 (PM2, PM15).

2. Do not set the pin that is set by ADPC as digital I/O by ADS.

3. Be sure to first set the ADCEN bit of peripheral enable register 0 (PER0) to 1 when setting up the ADPC register. If ADCEN = 0, writing to the ADPC register is ignored and specified values are returned to the initial values.

Remark P20/ANI0 to P27/ANI7, P150/ANI8 to P153/ANI11: 78K0R/KF3-C
P20/ANI0 to P27/ANI7, P150/ANI8 to P157/ANI15: 78K0R/KG3-C

(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 10-11. Formats of Port Mode Registers 2 and 15 (PM2, PM15)

Address: FFF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

Address: FFF2FH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM15	PM157	PM156	PM155	PM154	PM153	PM152	PM151	PM150

PMmn	Pmn pin I/O mode selection (mn = 20 to 27, 150 to 157)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark P150 to P153: 78K0R/KF3-C
P150 to P157: 78K0R/KG3-C

The 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 10-3. 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.	

Remark P20/ANI0 to P27/ANI7, P150/ANI8 to P153/ANI11: 78K0R/KF3-C
P20/ANI0 to P27/ANI7, P150/ANI8 to P157/ANI15: 78K0R/KG3-C

10.4 A/D Converter Operations

10.4.1 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 the A/D conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM, and set the operation mode by using bit 6 (ADMD) of ADM.

<3> Set bit 0 (ADCE) of A/D converter mode register (ADM) to 1 to start the operation of the A/D voltage comparator.

<4> Set the 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 (PM2 and PM15).

<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.
A timer trigger wait state is entered if the timer trigger mode is set in step <7>.
(<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_{REF} by the tap selector.

<10> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, 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_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and sampled voltage are compared and bit 8 of SAR is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<12> Comparison is continued in this way up to bit 0 of SAR.

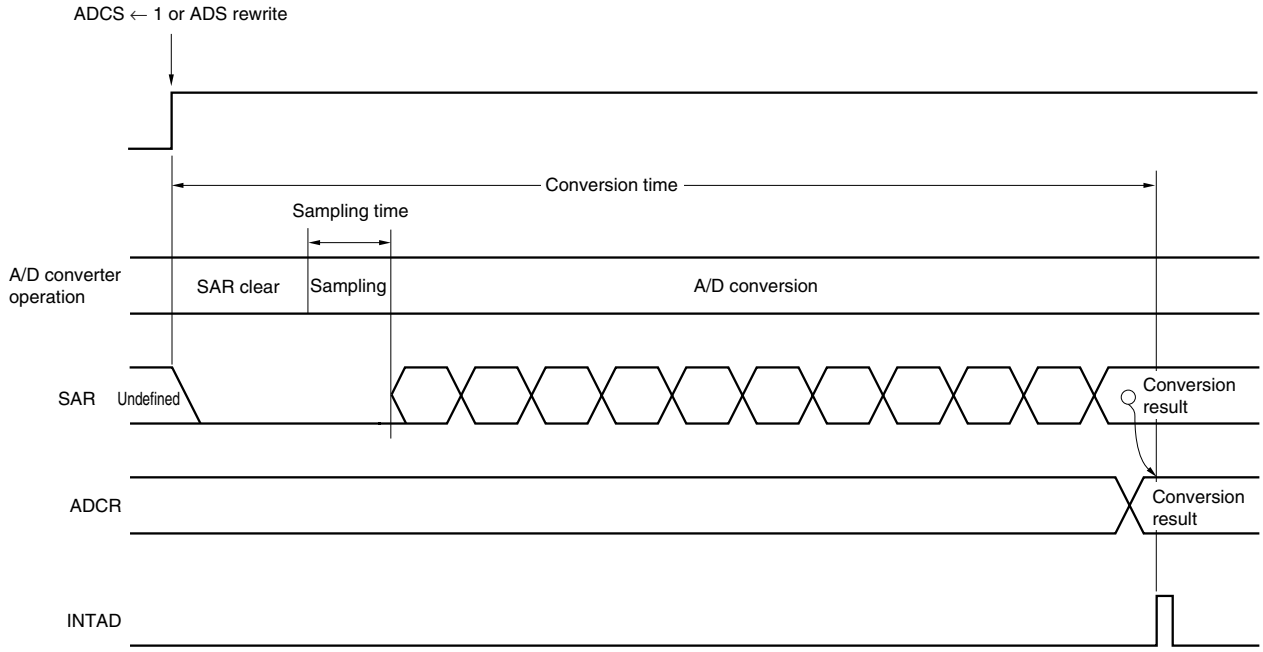
<13> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<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 μs or longer, and start <6>. To change a channel of A/D conversion, start from <5>.

Caution Make sure the period of <3> to <6> is 1 μs 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 10-12. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS) of 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.

10.4.2 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.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

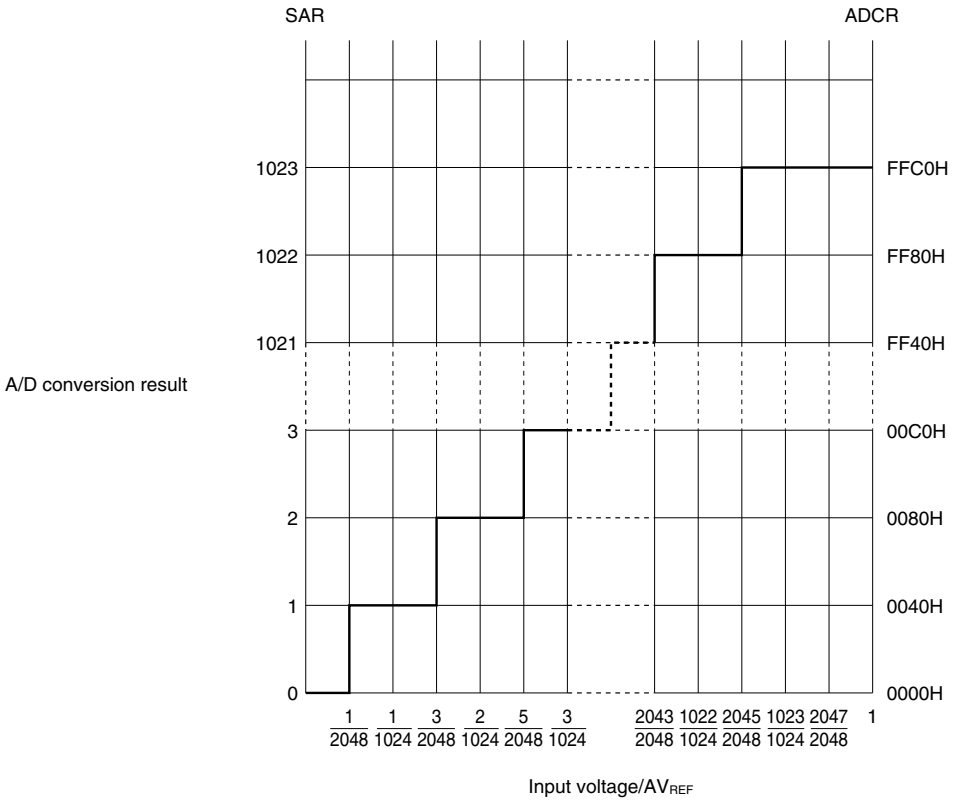
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 10-13 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 10-13. Relationship Between Analog Input Voltage and A/D Conversion Result



Remark ANI0 to ANI11: 78K0R/KF3-C
ANI0 to ANI15: 78K0R/KG3-C

10.4.3 A/D converter operation modes

The select mode and scan mode are provided as the A/D converter operation modes.

(1) Select mode

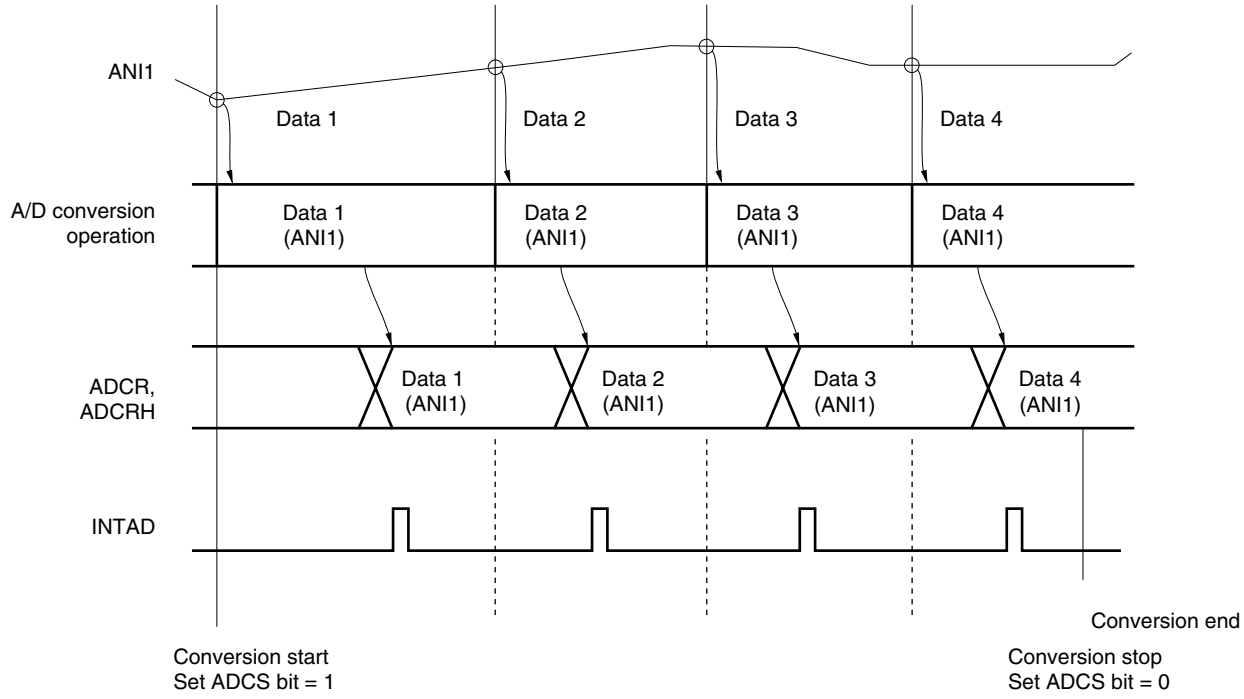
One analog input specified by the analog input channel specification register (ADS), while the ADMD bit of A/D converter mode register (ADM) is 0, is A/D converted.

When A/D conversion is complete, the conversion result is stored in the A/D conversion result register (ADCR) and the A/D conversion end interrupt request signal (INTAD) is generated.

After A/D conversion has been completed, A/D conversion is repeated successively, unless the ADCS bit is set to 0.

If anything is written to ADM or ADS during conversion, A/D conversion is aborted. In this case, A/D conversion is started again from the beginning.

Figure 10-14. Example of Select Mode Operation Timing



(2) Scan mode

The four analog input channels of scans 0 to 3, which are specified by the analog input channel specification register (ADS), while the ADMD bit of A/D converter mode register (ADM) is 1, are A/D converted successively. A/D conversion is performed in sequence, starting from the analog input channel specified by scan 0.

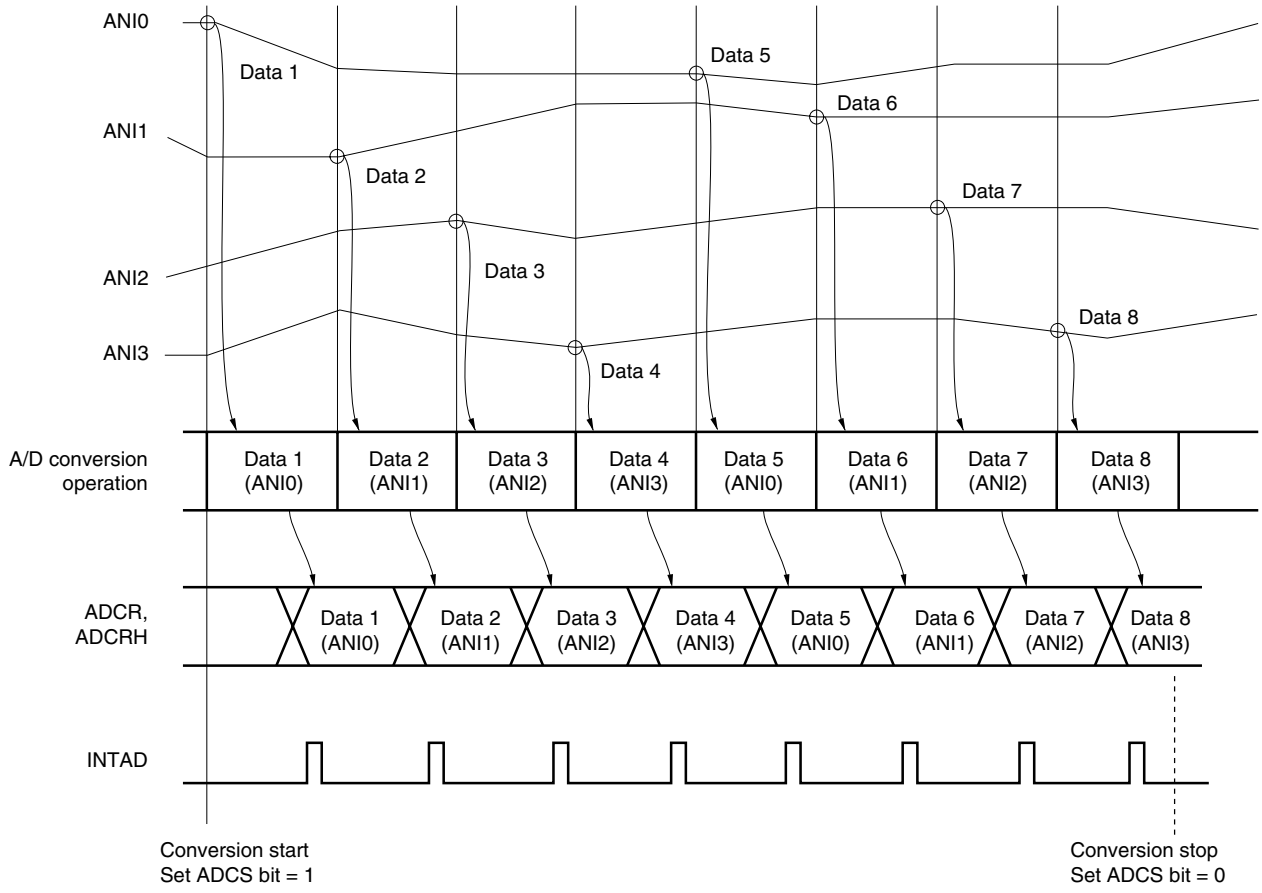
When A/D conversion of one analog input is complete, the conversion result is stored in the A/D conversion result register (ADCR) and the A/D conversion end interrupt request signal (INTAD) is generated.

The A/D conversion results of all the analog input channels are stored in ADCR. It is therefore recommended to save the contents of ADCR to RAM, once A/D conversion of one analog input channel has been completed.

When one A/D conversion ends, the next A/D conversion is started successively, until ADCS bit is set to 0.

If anything is written to ADM or ADS during conversion, A/D conversion is aborted. In this case, A/D conversion is started again from the analog input channel of scan 0.

Figure 10-15. Example of Scan Mode Operation Timing



The setting methods are described below.

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1.
<2> Select the conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM, and select the operation mode by using bit 6 (ADMD) of ADM.
<3> Set bit 0 (ADCE) of A/D converter mode register (ADM) to 1.
<4> 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), 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).
<5> Select a channel to be used by using bits 3 to 0 (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 3 to 0 (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) to 0.

Cautions 1. Make sure the period of <3> to <6> is 1 μs or more.
2. <3> may be done between <4> and <5>.
3. <3> can be omitted. However, ignore data of the first conversion after <7> in this case.
4. 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.

Remark P153 to P150: 78K0R/KF3-C
P157 to P150: 78K0R/KG3-C

10.5 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.
Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.
Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 10-16. Overall Error

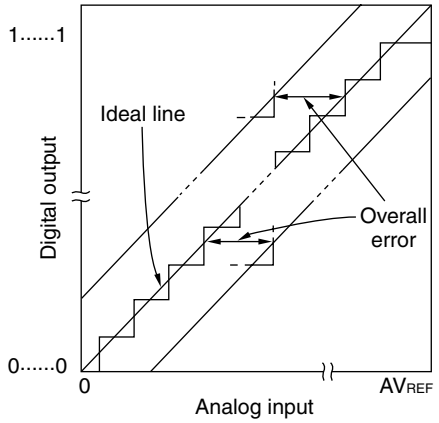
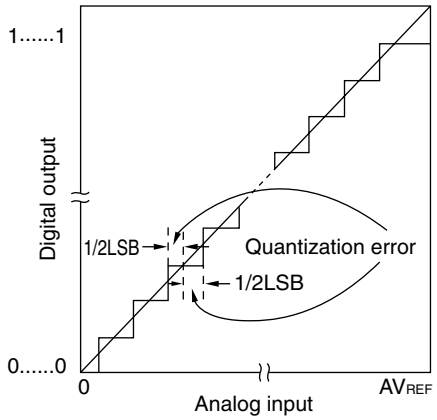


Figure 10-17. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.
If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

This error indicates basic A/D conversion characteristics if the voltage applied to the analog input pins of a channel is monotonically increases from AV_{SS} to AV_{REF0} little by little. If the input voltage increases or decreases or if multiple channels are used, see **10.5 (2) Overall error**.

Figure 10-18. Zero-Scale Error

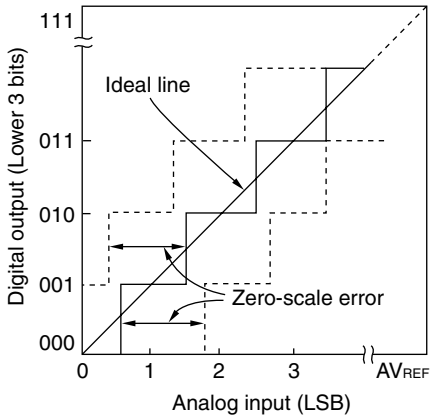


Figure 10-19. Full-Scale Error

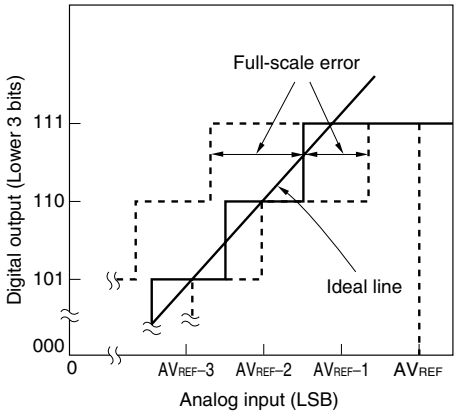


Figure 10-20. Integral Linearity Error

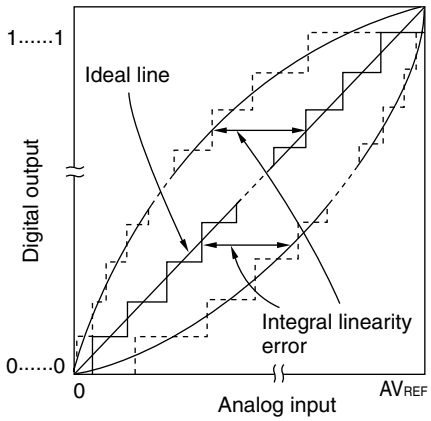
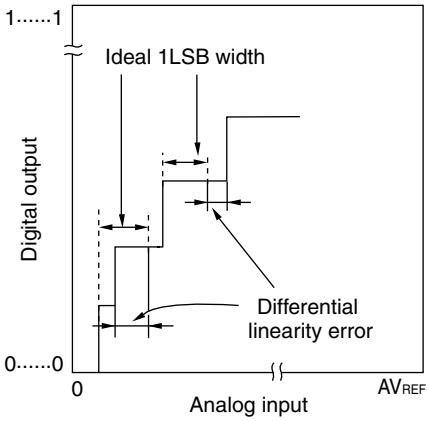


Figure 10-21. Differential Linearity Error



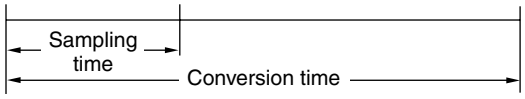
(8) Conversion time

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.



10.6 Cautions for A/D Converter

(1) Operating current in STOP mode

Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 0). The operating current can be reduced by setting 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

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_{REF}, while the A/D converter is stopped.

(3) Input range of ANI0 to ANI15

Observe the rated range of the ANI0 to ANI15 input voltage. If a voltage of AV_{REF} 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_{REF} pin and pins ANI0 to ANI15.

<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

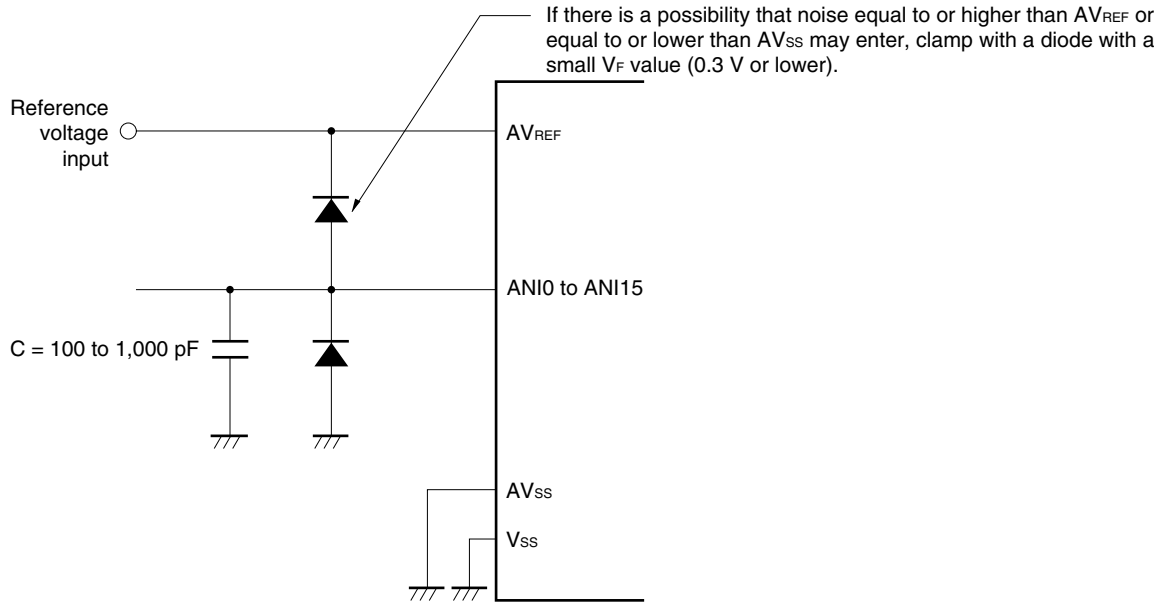
<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external C as shown in Figure 10-22 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Remark ANI0 to ANI11: 78K0R/KF3-C
ANI0 to ANI15: 78K0R/KG3-C

Figure 10-22. Analog Input Pin Connection



(6) ANI0/P20 to ANI7/P27 and ANI8/P150 to ANI15/P157

<1> The analog input pins (ANI0 to ANI15) are also used as input port pins (P20 to P27, 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 AVREF.

<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 1 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI15 pins (see **Figure 10-22**).

(8) AVREF pin input impedance

A series resistor string of several tens of kΩ is connected between the AVREF and AVSS 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 AVREF and AVSS pins, resulting in a large reference voltage error.

Remark P20/ANI0 to P27/ANI7, P150/ANI8 to P153/ANI11: 78K0R/KF3-C
P20/ANI0 to P27/ANI7, P150/ANI8 to P157/ANI15: 78K0R/KG3-C

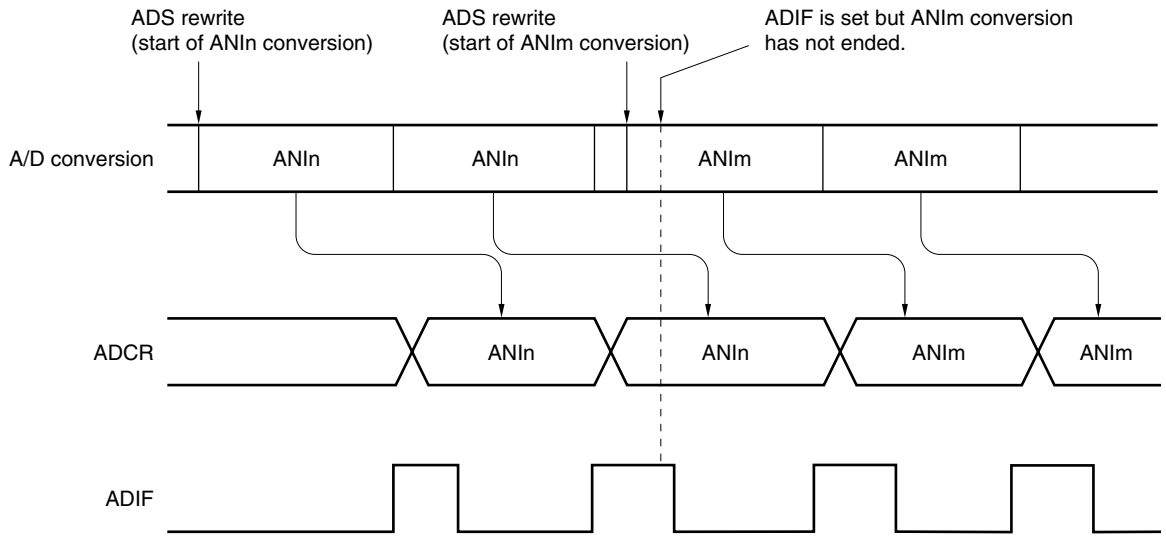
(9) Interrupt request flag (ADIF)

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 10-23. Timing of A/D Conversion End Interrupt Request Generation



Remark n = 0 to 11, m = 0 to 11: 78K0R/KF3-C
n = 0 to 15, m = 0 to 15: 78K0R/KG3-C

(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 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.

(11) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to 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, or 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 10-24. Internal Equivalent Circuit of ANIn Pin

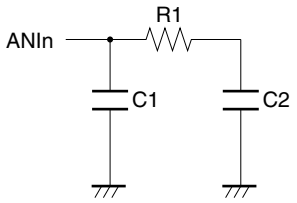


Table 10-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	Mode	R1	C1	C2
4.0 V ≤ V_{DD} ≤ 5.5 V	Normal	5.2 kΩ	8 pF	6.3 pF
	High speed 1	5.2 kΩ		
	High speed 2	7.8 kΩ		
2.7 V ≤ V_{DD} < 4.0 V	Normal	18.6 kΩ		
	High speed 2	7.8 kΩ		

Remarks 1. The resistance and capacitance values shown in Table 10-4 are not guaranteed values.

2. 78K0R/KF3-C: n = 0 to 11

78K0R/KG3-C: n = 0 to 15

(13) Starting the A/D converter

Start the A/D converter after the AV_{REF} voltages stabilize.

CHAPTER 11 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^2C) in combination.

Function assignment of each channel supported by the 78K0R/Kx3-C is as shown below.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	–
	1	CSI01		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–

(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 can be used.

11.1 Functions of Serial Array Unit

Each serial interface supported by the 78K0R/Kx3-C has the following features.

11.1.1 3-wire serial I/O (CSI00, CSI01, CSI10, CSI20)

This is a clocked communication function that uses three lines: serial clock (f_{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
- Maximum transfer rate

  During master communication: Max. $f_{CLK}/4$, during slave communication: Max. $f_{MCK}/6$ Note

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Note Use the clocks within a range satisfying the $\overline{SCK}$ cycle time (t_{KCY}) characteristics (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

11.1.2 UART (UART0, UART1, UART2)

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

11.1.3 Simplified I²C (IIC10, IIC20)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master 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[Note] and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Parity error (ACK error)

* [Functions not supported by simplified I²C]

- 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 **11.7.3 (2) Processing flow** for details.

Remarks 1. To use the full-function I²C bus, see **CHAPTER 12 SERIAL INTERFACE IICA**.

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

11.2 Configuration of Serial Array Unit

Serial array unit includes the following hardware.

Table 11-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits
Buffer register	Lower 8 bits of serial data register mn (SDRmn)[Note]
Serial clock I/O	SCK00, SCK01, SCK10, SCK20 pins (for 3-wire serial I/O), SCL10, SCL20 pins (for simplified I^2C)
Serial data input	SI00, SI01, SI10, SI20 pins (for 3-wire serial I/O), RxD0, RxD1, RxD2 pins (for UART)
Serial data output	SO00, SO01, SO10, SO20 pins (for 3-wire serial I/O), TxD0, TxD1, TxD2 pins (for UART)
Serial data I/O	SDA10, SDA20 pins (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn) • Port input mode registers 0, 1, 14 (PIM0, PIM1, PIM14) • Port output mode registers 0, 1, 14 (POM0, POM1, POM14) • Port mode registers 0, 1, 4, 14 (PM0, PM1, PM4, PM14) • Port registers 0, 1, 4, 14 (P0, P1, P4, P14)

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), mn = 00 to 03, 10, 11
p: CSI number (p = 00, 01, 10, 20), q: UART number (q = 0 to 2), r: IIC number (r = 10, 20)

Figure 11-1 shows the block diagram of serial array unit 0.

Figure 11-1. Block Diagram of Serial Array Unit 0

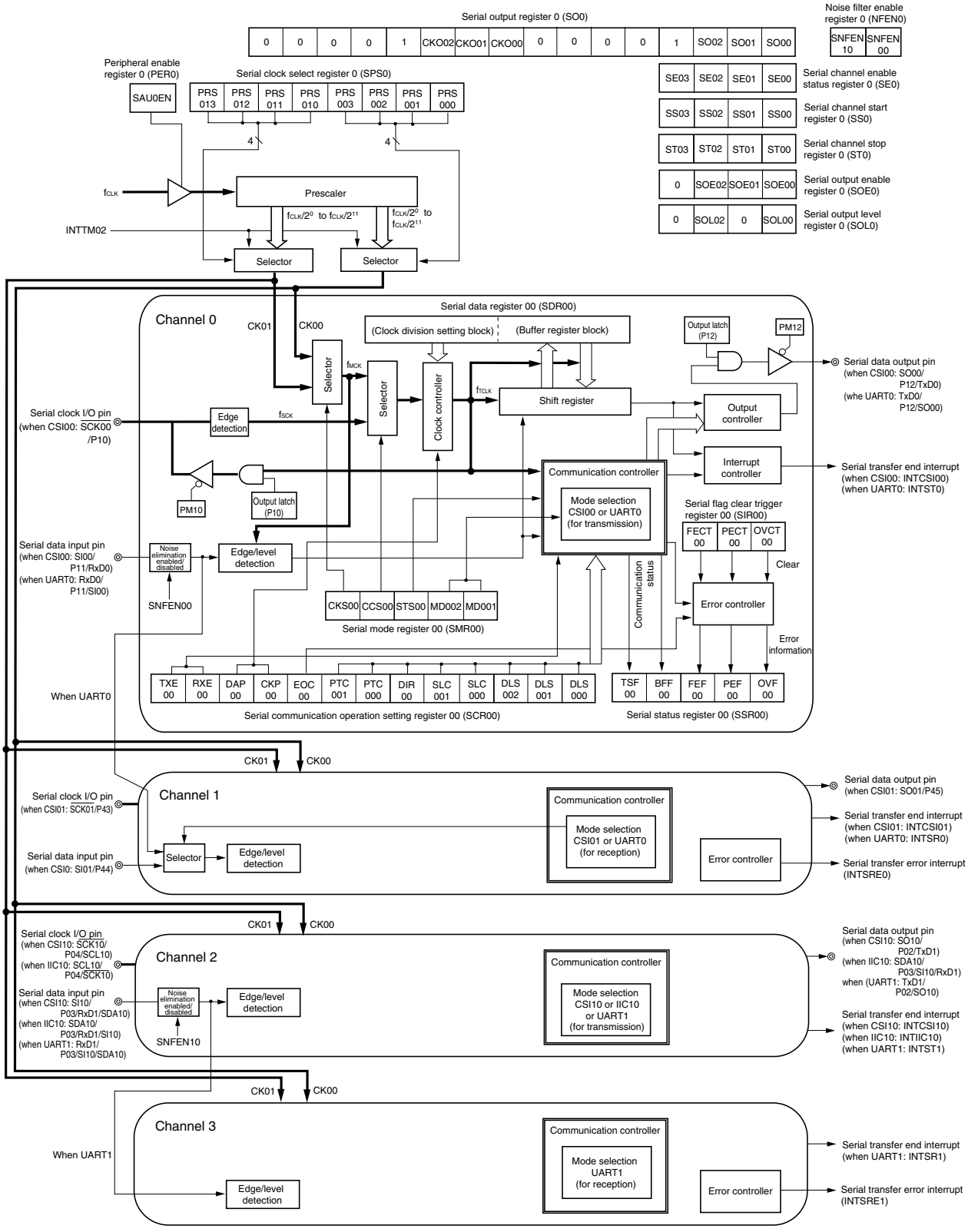
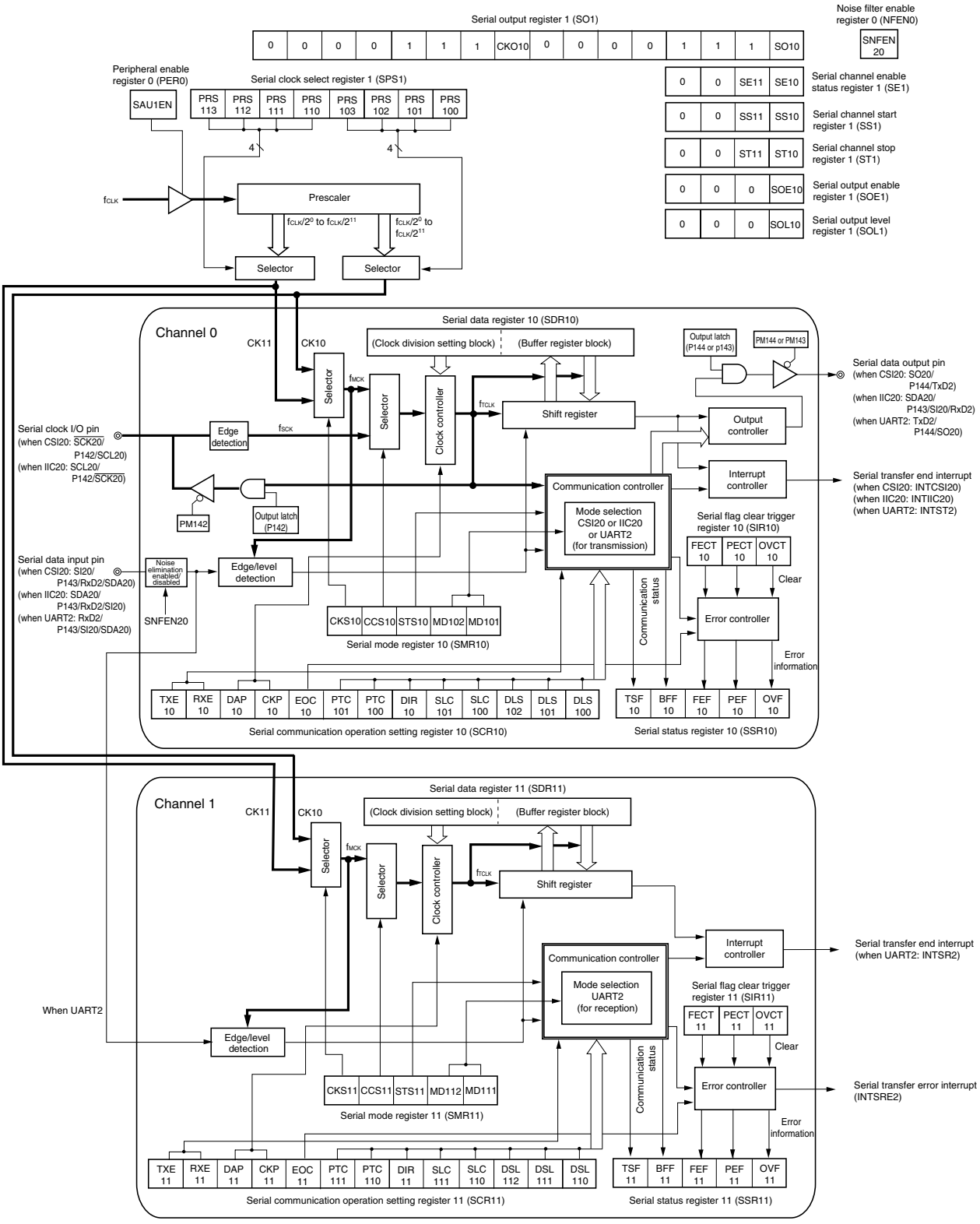


Figure 11-2 shows the block diagram of serial array unit 1.

Figure 11-2. Block Diagram of Serial Array Unit 1



(1) Shift register

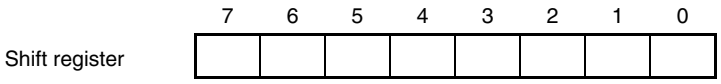
This is an 8-bit register that converts parallel data into serial data or vice versa.

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 (f_{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[Note] as the following SFR, depending on the communication mode.

- CSIp communication … SIOp (CSIp data register)
- UARTq reception … RXDq (UARTq receive data register)
- UARTq transmission … TXDq (UARTq transmit data register)
- IICr communication … SIOr (IICr data register)

Reset signal generation clears this register to 0000H.

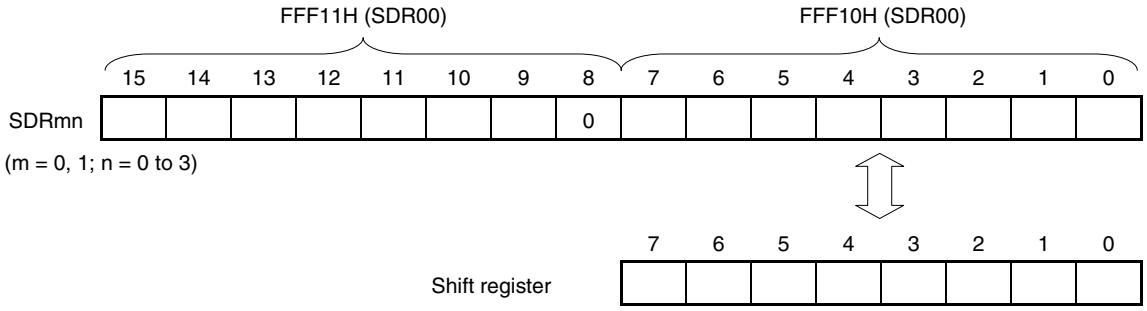
Note Writing in 8-bit units is prohibited when the operation is stopped (SEmn = 0).

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), mn = 00 to 03, 10, 11, p: CSI number (p = 00, 01, 10, 20), q: UART number (q = 0 to 2), r: IIC number (r = 10, 20)

Figure 11-3. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03), FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11) After reset: 0000H R/W



Caution Be sure to clear bit 8 to "0".

Remarks 1. For the function of the higher 7 bits of SDRmn, see **11.3 Registers Controlling Serial Array Unit**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11,

11.3 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)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 0, 1, 14 (PIM0, PIM1, PIM14)
- Port output mode registers 0, 1, 14 (POM0, POM1, POM14)
- Port mode registers 0, 1, 4, 14 (PM0, PM1, PM4, PM14)
- Port registers 0, 1, 4, 14 (P0, P1, P4, P14)

Remark m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 3)
mn = 00 to 03, 10, 11

(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 11-4. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

SAUmEN	Control of serial array unit m input clock
0	Stops supply of input clock. • SFR used by serial array unit m cannot be written. • Serial array unit m is in the reset status.
1	Supplies input clock. • SFR used by serial array unit m can be read/written.

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 noise filter enable register (NFEN0), port input mode registers (PIM0, PIM1, PIM14), port output mode registers (POM0, POM1, POM14), port mode registers (PM0, PM1, PM4, PM14), and port registers (P0, P1, P4, P14)).

2. After setting bit SAUmEN of the PER0 register to 1, be sure to set SPSm register after 4 or more f_{CLK} clocks have elapsed.

3. Be sure to clear bit 6 of the PER0 register to 0.

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 11-5. 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	Section of operation clock (CKmp)[Note 1]				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	1	1	INTTM02 if m = 0[Note 2], setting prohibited if m = 1				
Other than above				Setting prohibited				

Notes 1. When changing the clock selected for f_{CLK} (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 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAU0 can be used at the fixed division ratio of the subsystem clock regardless of the f_{CLK} frequency by setting TAU0 and SAU0 as follows.

<TAU0> Select $f_{SUB}/4$ as the input clock of channel 2 of TAU0. (Set TIS02 to 1.)

<SAU0> Select INTTM02 by using the SPS0 register.

However, when changing f_{CLK}, SAU0 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 bit SAUmEN of the PER0 register to 1, be sure to set SPSm register after 4 or more f_{CLK} clocks have elapsed.

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

f_{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 (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, an operation mode (CSI, UART, or I^2C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting 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 11-6. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), F0150H, F0151H (SMR10), F0152H, F0153H (SMR11) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn	0	SIS mn0	1	0	0	MD mn2	MD mn1	MD mn0

CKS mn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock f_{MCK} is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

CCS mn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by CKSmn bit
1	Clock input from SCK pin (slave transfer in CSI mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of f_{MCK} is set by the higher 7 bits of the SDRmn register.	

STS mn	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I^2C).
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), mn = 00 to 03, 10, 11

Figure 11-6. 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)

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^2C mode
1	1	Setting prohibited

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting MDmn0 to 1 when SDRmn data has run out.	

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), mn = 00 to 03, 10, 11

(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.

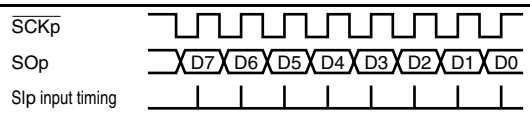
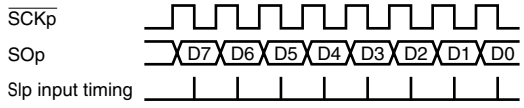
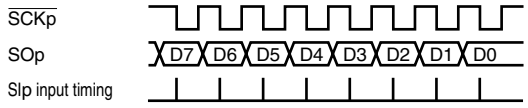
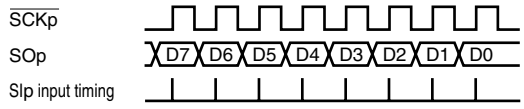
Reset signal generation sets this register to 0087H.

Figure 11-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F0158H, F0159H (SCR10), F015AH, F015BH (SCR11) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	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		1
0	1		2
1	0		3
1	1		4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C 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), mn = 00 to 03, 10, 11,
p: CSI number (p = 00, 01, 10, 20)

Figure 11-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F0158H, F0159H (SCR10), F015AH, F015BH (SCR11) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	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^2C mode, and during UART transmission[Note 1]. 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[Note 2].	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the CSI mode and simplified I^2C 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^2C mode.	

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^2C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the CSI mode.		

Notes 1. When not using CSI01 with EOC01 = 0, error interrupt INTSRE0 may be generated.

2. 0 is always added regardless of the data contents.

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), mn = 00 to 03, 10, 11

Figure 11-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (3/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F0158H, F0159H (SCR10), F015AH, F015BH (SCR11) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

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^2C 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), mn = 00 to 03, 10, 11

(5) Higher 7 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 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 (f_{MCK}).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by the higher 7 bits of the SDRmn register is used as the transfer clock.

The lower 8 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8 bits.

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 11-8. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03), FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11) After reset: 0000H R/W

	FFF11H (SDR00)								FFF10H (SDR00)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0								

SDRmn[15:9]							Transfer clock setting by dividing the operating clock (f_{MCK})
0	0	0	0	0	0	0	f_{MCK}/2
0	0	0	0	0	0	1	f_{MCK}/4
0	0	0	0	0	1	0	f_{MCK}/6
0	0	0	0	0	1	1	f_{MCK}/8
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	f_{MCK}/254
1	1	1	1	1	1	1	f_{MCK}/256

Cautions 1. Be sure to clear bit 8 to "0".

2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

3. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I^2C is used. Setting SDRmn[15:9] = 0000001B or more.

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 **11.2 Configuration of Serial Array Unit**.

2. m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 3)
mn = 00 to 03, 10, 11

(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 11-9. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), F0140H, F0141H (SSR10), F0142H, F0143H (SSR11) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn	PEF mn	OVF mn

TSF mn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

BFF mn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
<Clear conditions> • Transferring transmit data from the SDRmn register to the shift register ends during transmission. • Reading receive data from the SDRmn register ends during reception. • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode). • Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 11-9. 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)

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	An error occurs (during UART reception).

<Clear condition>
- 1 is written to the FECTmn bit of the SIRmn register.

<Set condition>
- A stop bit is not detected when UART reception ends.

PEF mn	Parity error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception) or ACK is not detected (during I^2C transmission).

<Clear condition>
- 1 is written to the PECTmn bit of the SIRmn register.

<Set condition>
- The parity of the transmit data and the parity bit do not match when UART reception ends (parity error).
- No ACK signal is returned from the slave channel at the ACK reception timing during I^2C transmission (ACK is not detected).

OVF mn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs

<Clear condition>
- 1 is written to the OVCTmn bit of the SIRmn register.

<Set condition>
- Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode).
- Transmit data is not ready for slave transmission or transmission and reception in CSI mode.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

(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 11-10. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03), F0148H, F0149H (SIR10), F014AH, F014BH (SIR11) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FEC Tmn	PEC Tmn	OVC Tmn

FEC Tmn	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PEC Tmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVC Tmn	Clear trigger of overrun error flag of channel n
0	Not cleared
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), mn = 00 to 03, 10, 11

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 11-11. 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	SEm3	SEm2	SEm1	SEm0

SEmn	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[Note]).
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), mn = 00 to 03, 10, 11

(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 11-12. 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	SSm3	SSm2	SSm1	SSm0

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), mn = 00 to 03, 10, 11

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 11-13. 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	STm3	STm2	STm1	STm0

STmn	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[Note].)

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), mn = 00 to 03, 10, 11

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 11-14. 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	0	SOE02	SOE01	SOE00

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	0	0	SOE10

SOEmn	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 3 of SOE0, and bits 15 to 1 of SOE1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2),
mn = 00 to 02, 10

(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/$\overline{\text{SCK10}}$/SCL10, P10/$\overline{\text{SCK00}}$, P12/SO00/TxD0, P43/$\overline{\text{SCK01}}$, P45/SO01, P142/$\overline{\text{SCK20}}$/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 11-15. 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
SO0	0	0	0	0	1	CKO02	CKO01	CKO00	0	0	0	0	1	SO02	SO01	SO00

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
SO1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	1	1	SO10

CKOmn	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

SOmn	Serial data output of channel n
0	Serial data output value is “0”.
1	Serial data output value is “1”.

Caution Be sure to set bits 11 and 3 of SO0, and bits 11 to 9, and 3 to 1 of SO1 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 2), mn = 00-02, 10

(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^2C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting 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 11-16. 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), mn = 00-02, 10

(14) 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^2C communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is enabled, CPU/peripheral operating clock (f_{CLK}) 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 11-17. 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	0	0	SNFEN20	0	SNFEN10	0	SNFEN00

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 to 5, 3, and 1 to "0".

(15) Port input mode registers 0, 1, 14 (PIM0, PIM1, PIM14)

These registers set the input buffer of ports 0, 1, and 14 in 1-bit units.

PIM0, PIM1, and PIM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 11-18. Format of Port Input Mode Registers 0, 1, and 14 (PIM0, PIM1, 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 F0041H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIM1	0	0	0	0	0	0	PIM11	PIM10

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, 1, 14; n = 0 to 4)
0	Normal input buffer
1	TTL input buffer

(16) Port output mode registers 0, 1, 14 (POM0, POM1, POM14)

These registers set the output mode of ports 0, 1, and 14 in 1-bit units.

POM0, POM1, and POM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 11-19. Format of Port Output Mode Registers 0, 1, and 14 (POM0, POM1, 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 F0051H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
POM1	0	0	0	0	0	POM12	0	POM10

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, 1, 14; n = 0, 2 to 4)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

(17) Port mode registers 0, 1, 4, 14 (PM0, PM1, PM4, PM14)

These registers set input/output of ports 0, 1, 4 and 14 in 1-bit units.

When using the P02/SO10/TxD1, P03/SI10/RxD1/SDA10, P04/$\overline{\text{SCK10}}$/SCL10, P10/$\overline{\text{SCK00}}$, P12/SO00/TxD0, P43/$\overline{\text{SCK01}}$, P45/SO01, P142/$\overline{\text{SCK20}}$/SCL20, P143/SI20/RxD2/SDA20, and P144/SO20/TxD2 pins for serial data output or serial clock output, clear the PM02, PM03, PM04, PM10, PM12, PM43, PM45, PM142, PM143, and PM144 bits to 0, and set the output latches of P02, P03, P04, P10, P12, P43, P45, P142, P143, and P144 to 1.

When using the P03/SI10/RxD1/SDA10, P04/$\overline{\text{SCK10}}$/SCL10, P10/$\overline{\text{SCK00}}$, P11/SI00/RxD0, P43/$\overline{\text{SCK01}}$, P44/SI01, P142/$\overline{\text{SCK20}}$/SCL20, and P143/SI20/RxD2/SDA20 pins for serial data input or serial clock input, set the PM03, PM04, PM10, PM11, PM43, PM44, PM142, and PM143 bits to 1. At this time, the output latches of P03, P04, P10, P11, P43, P44, P142, and P143 may be 0 or 1.

PM0, PM1, PM4, and PM14 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 11-20. Format of Port Mode Registers 0, 1, 4, and 14 (PM0, PM1, PM4, PM14)

Address: FFF20H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM0	1	PM06	PM05	PM04	PM03	PM02	PM01 Note	PM00 Note

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: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	PM145 Note	PM144	PM143	PM142	PM141 Note	PM140

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

Note 78K0R/KF3-C only

11.4 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/$\overline{\text{SCK10}}$/SCL10, P10/$\overline{\text{SCK00}}$, P11/SI00/RxD0, P12/SO00/TxD0, P43/$\overline{\text{SCK01}}$, P44/SI01, P45/SO01, P142/$\overline{\text{SCK20}}$/SCL20, P143/SI20/SDA20/RxD2, or P144/SO20/TxD2 pin can be used as ordinary port pins in this mode.

11.4.1 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 11-21. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAUm to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	RTCEN		ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
	×	0	×	×	0/1	0/1	×	×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Cautions 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read (except for noise filter enable register (NFEN0), port input mode registers (PIM0, PIM1, PIM14), port output mode registers (POM0, POM1, POM14), port mode registers (PM0, PM1, PM4, PM14), and port registers (P0, P1, P4, P14)).

2. Be sure to clear bit 6 of the PER0 register to 0.

Remark m: Unit number (m = 0, 1), ▢: Setting disabled (fixed by hardware)
×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

11.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 11-22. 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.

SEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SEm3 0/1	SEm2 0/1	SEm1 0/1	SEm0 0/1

0: Operation stops

* The SEm register is a read-only status register, whose operation is stopped by using the STm register.
With a channel whose operation is stopped, the value of CKOmn 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.

STm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	STm3 0/1	STm2 0/1	STm1 0/1	STm0 0/1

1: Clears SEmn to 0 and stops the communication operation

* Because STmn is a trigger bit, it is cleared immediately when SEmn = 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.

SOE0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	SOE01 0/1	SOE00 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SO0n value of the SO0 register can be set by software.

SOE1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE10 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SO10 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 11-22. 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.

SO0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	CKO02 0/1	CKO01 0/1	CKO00 0/1	0	0	0	0	1	SO02 0/1	SO01 0/1	SO00 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”.

SO1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	CKO10 0/1	0	0	0	0	1	1	1	SO10 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 CKO10 and SO10 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

11.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) Communication

This is a clocked communication function that uses three lines: serial clock (f_{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
- Maximum transfer rate

  During master communication: Max. $f_{CLK}/4$, during slave communication: Max. $f_{MCK}/6$ Note

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Note Use the clocks within a range satisfying the $\overline{SCK}$ cycle time (t_{KCY}) characteristics (see **CHAPTER 29 ELECTRICAL SPECIFICATION**).

The channels supporting 3-wire serial I/O (CSI00, CSI01, CSI10, CSI20) are channels 0 to 2 of SAU0 and channel 0 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	–
	1	CSI01		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–

3-wire serial I/O (CSI00, CSI01, CIS10, CSI20) performs the following six types of communication operations.

- Master transmission (See **11.5.1**.)
- Master reception (See **11.5.2**.)
- Master transmission/reception (See **11.5.3**.)
- Slave transmission (See **11.5.4**.)
- Slave reception (See **11.5.5**.)
- Slave transmission/reception (See **11.5.6**.)

11.5.1 Master transmission

Master transmission is an operation in which the 78K0R/Kx3-C outputs a transfer clock and transmits data to another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	$\overline{\text{SCK00}}$, SO00	$\overline{\text{SCK01}}$, SO01	$\overline{\text{SCK10}}$, SO10	$\overline{\text{SCK20}}$, SO20
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	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]Note f_{CLK}: System clock frequency			
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse			
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**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10

(1) Register setting

Figure 11-23. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 Note 0/1	CKOm1 Note 0/1	CKOm0 0/1	0	0	0	0	1	SOm2 Note 0/1	SOm1 Note 0/1	SOm0 0/1

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.

0: Serial data output value is "0"
1: Serial data output value is "1"

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 Note 0/1	SOEm1 Note 0/1	SOEm0 0/1

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 Note 0/1	SSm1 Note 0/1	SSm0 0/1

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISm0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

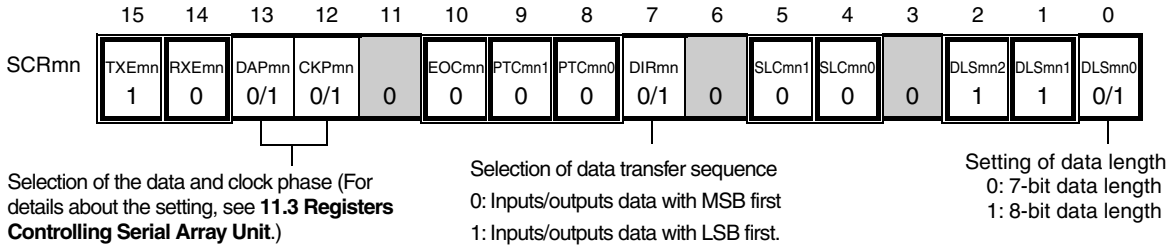
Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

Note Serial array unit 0 only.

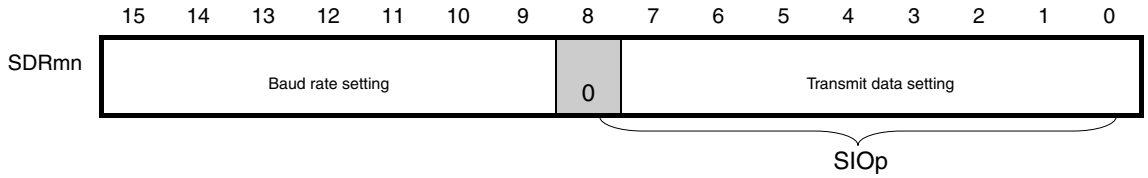
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)
□: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-23. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) (2/2)

(e) Serial communication operation setting register mn (SCRmn)



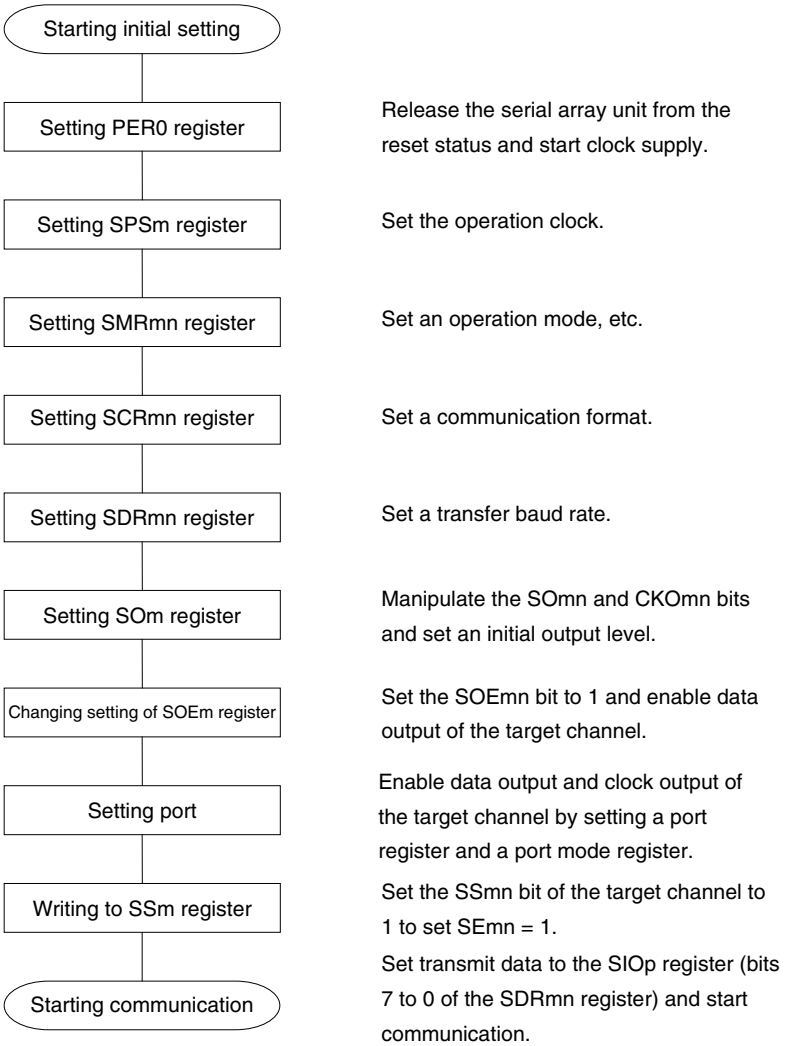
(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)
□: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

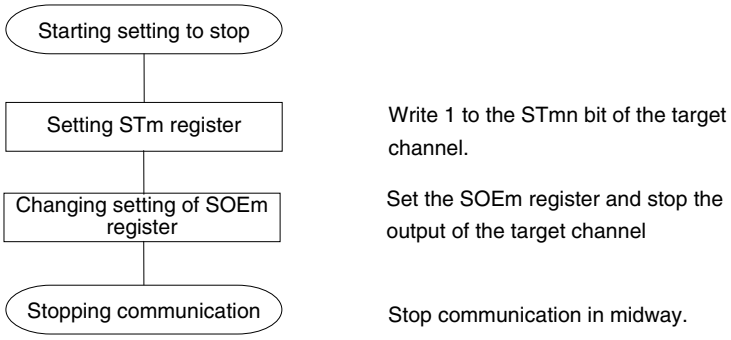
(2) Operation procedure

Figure 11-24. Initial Setting Procedure for Master Transmission



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

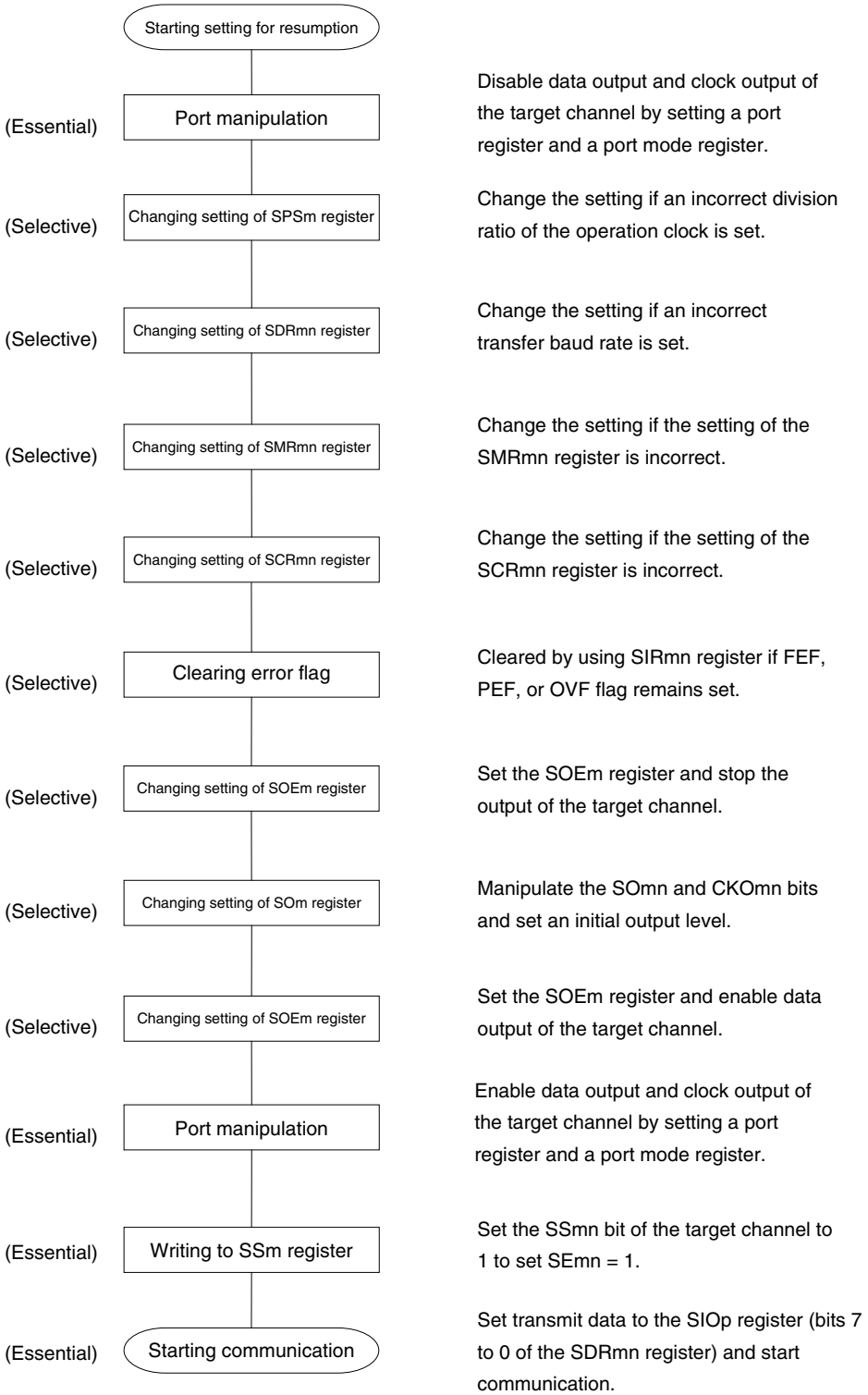
Figure 11-25. Procedure for Stopping Master Transmission



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-26 Procedure for Resuming Master Transmission**).

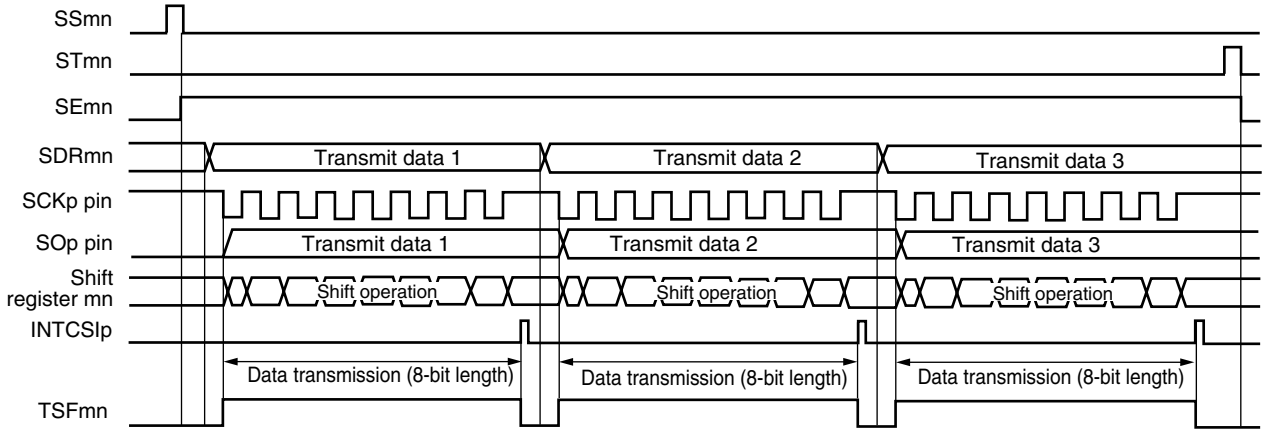
2. p: CSI number (p = 00, 01, 10, 20)

Figure 11-26. Procedure for Resuming Master Transmission



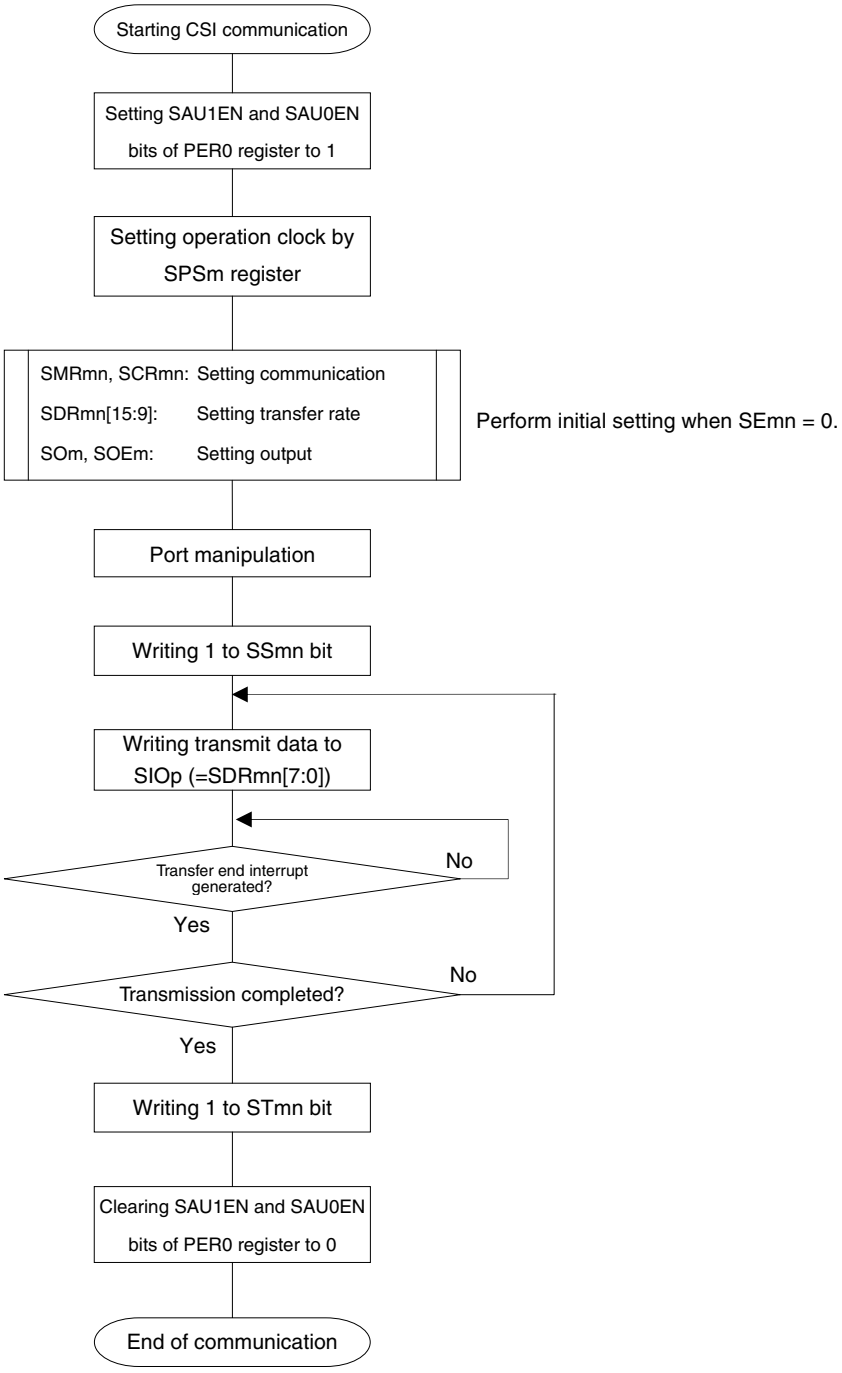
(3) Processing flow (in single-transmission mode)

Figure 11-27. 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 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)

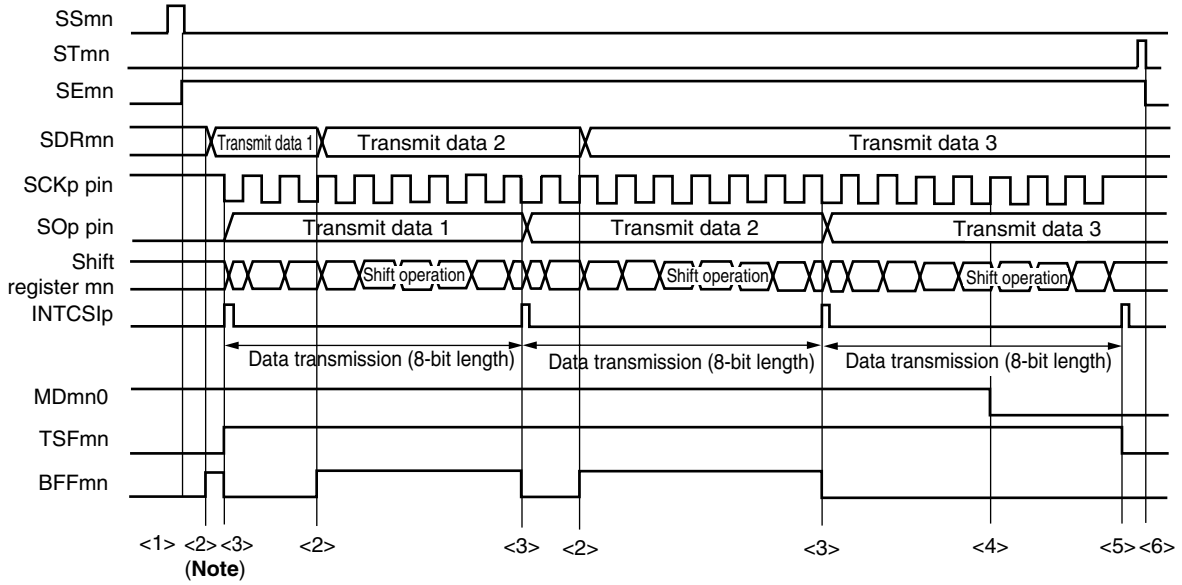
Figure 11-28. Flowchart of Master Transmission (in Single-Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 11-29. Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

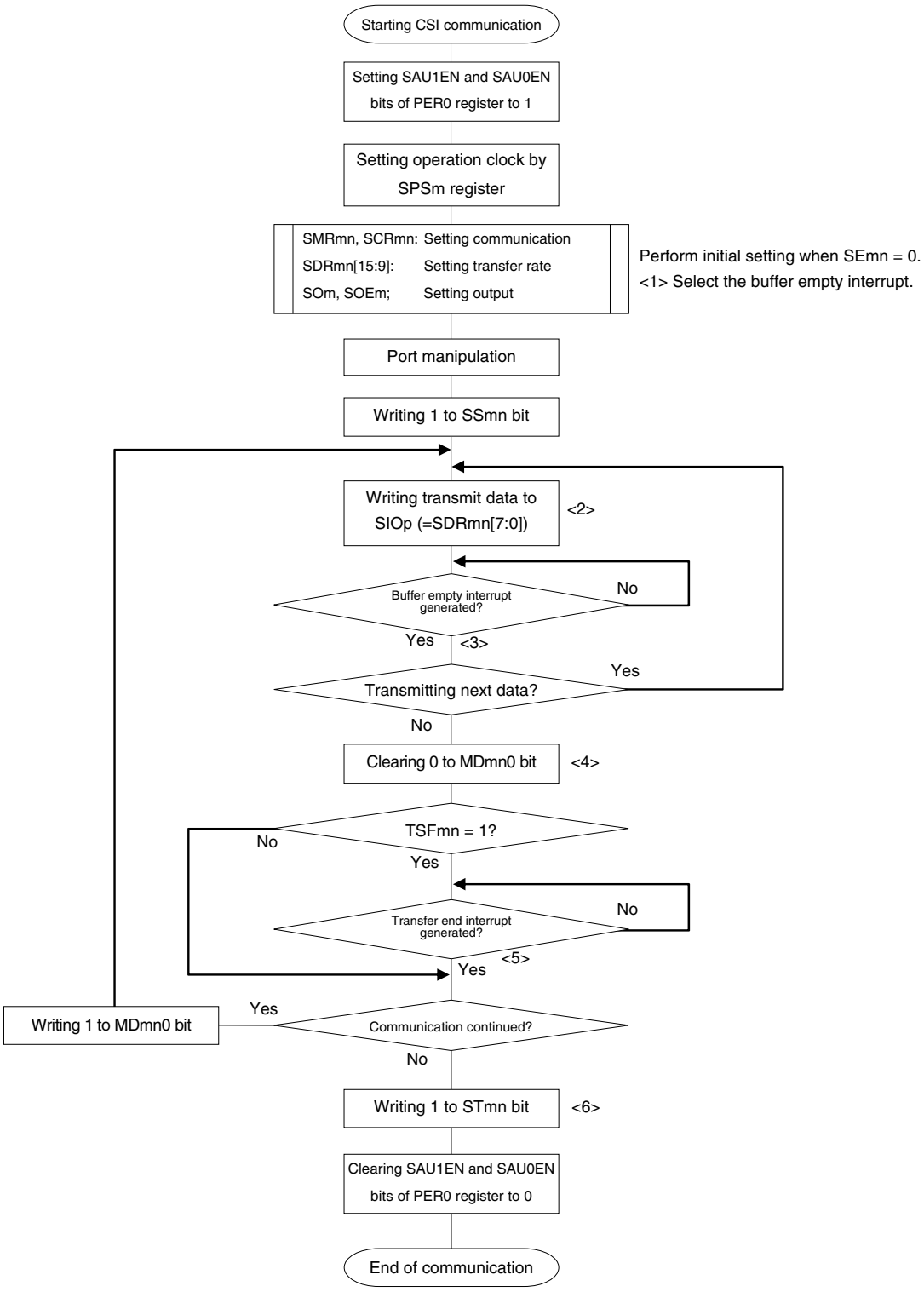


Note When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

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 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)

Figure 11-30. Flowchart of Master Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-29 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

11.5.2 Master reception

Master reception is an operation in which the 78K0R/Kx3-C outputs a transfer clock and receives data from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	$\overline{\text{SCK00}}$, SI00	$\overline{\text{SCK01}}$, SI01	$\overline{\text{SCK10}}$, SI10	$\overline{\text{SCK20}}$, SI20
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	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]Note f_{CLK}: System clock frequency			
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse			
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**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10

(1) Register setting

Figure 11-31. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 Note 0/1	CKOm1 Note 0/1	CKOm0 0/1	0	0	0	0	1	SOm2 ×	SOm1 ×	SOm0 ×

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.

(b) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 ×	SOEm0 ×

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 Note 0/1	SSm1 Note 0/1	SSm0 0/1

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISm0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

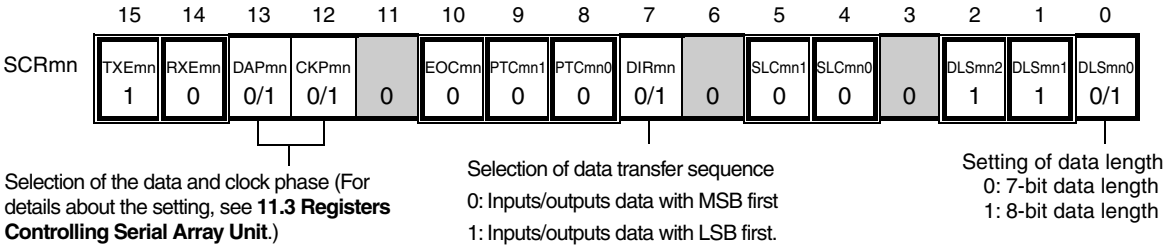
Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

Note Serial array unit 0 only.

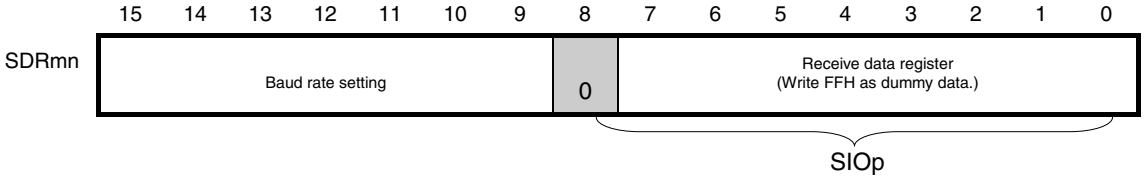
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)
□: Setting is fixed in the CSI master reception mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-31. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) (2/2)

(e) Serial communication operation setting register mn (SCRmn)



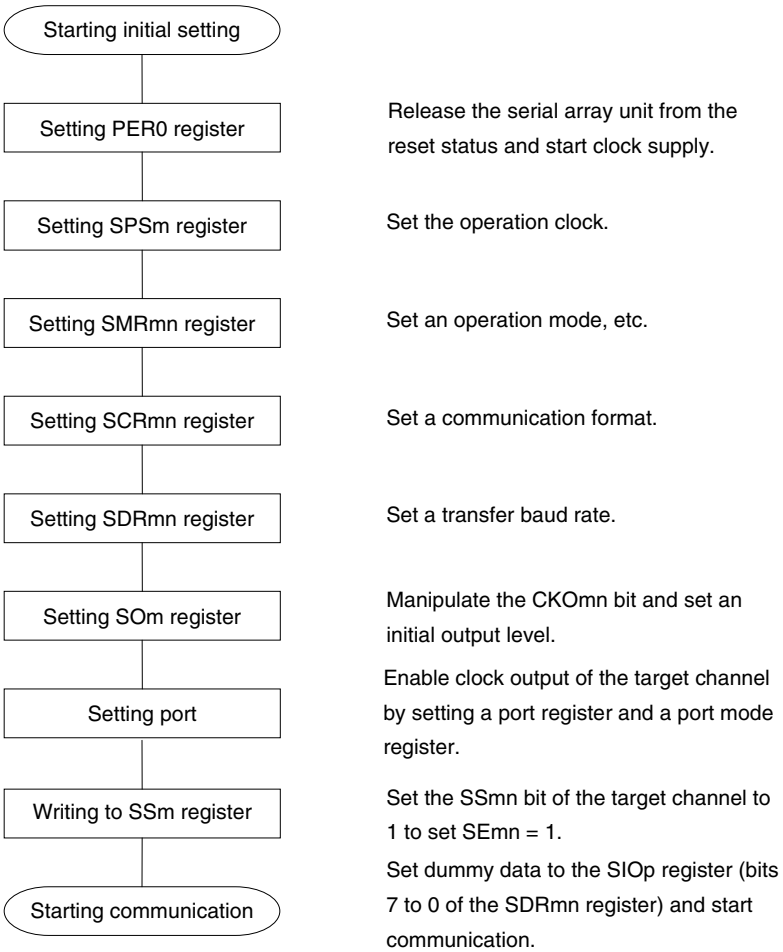
(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)
□: 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

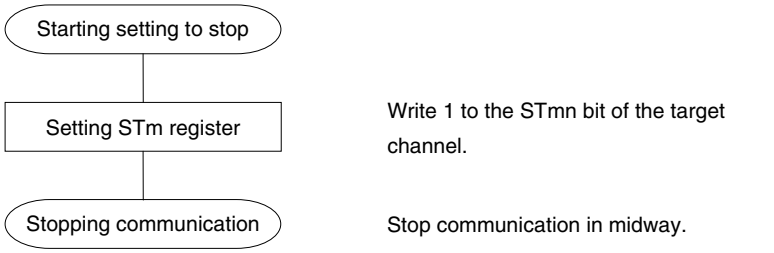
(2) Operation procedure

Figure 11-32. Initial Setting Procedure for Master Reception



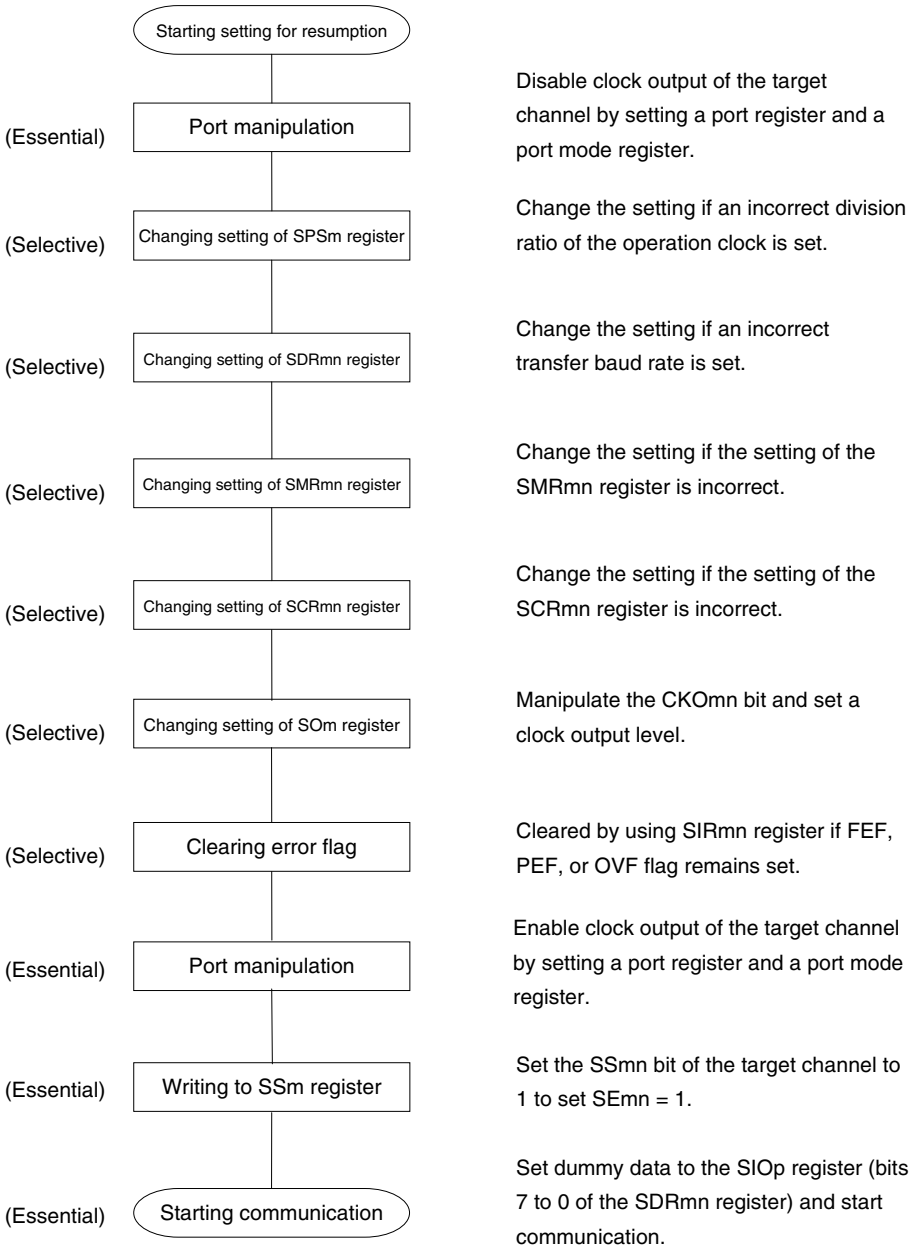
Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 11-33. Procedure for Stopping Master Reception



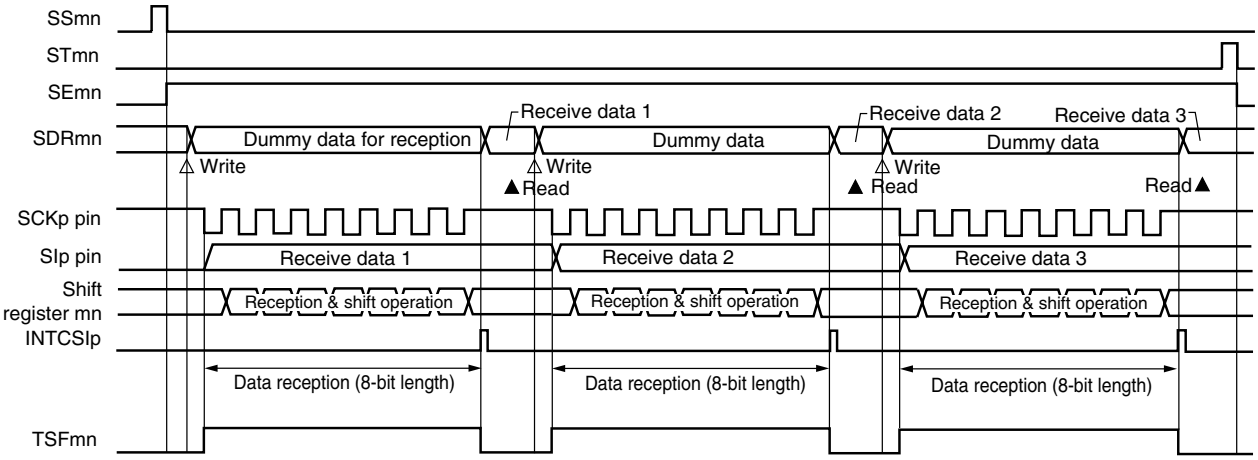
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-34 Procedure for Resuming Master Reception**).

Figure 11-34. Procedure for Resuming Master Reception



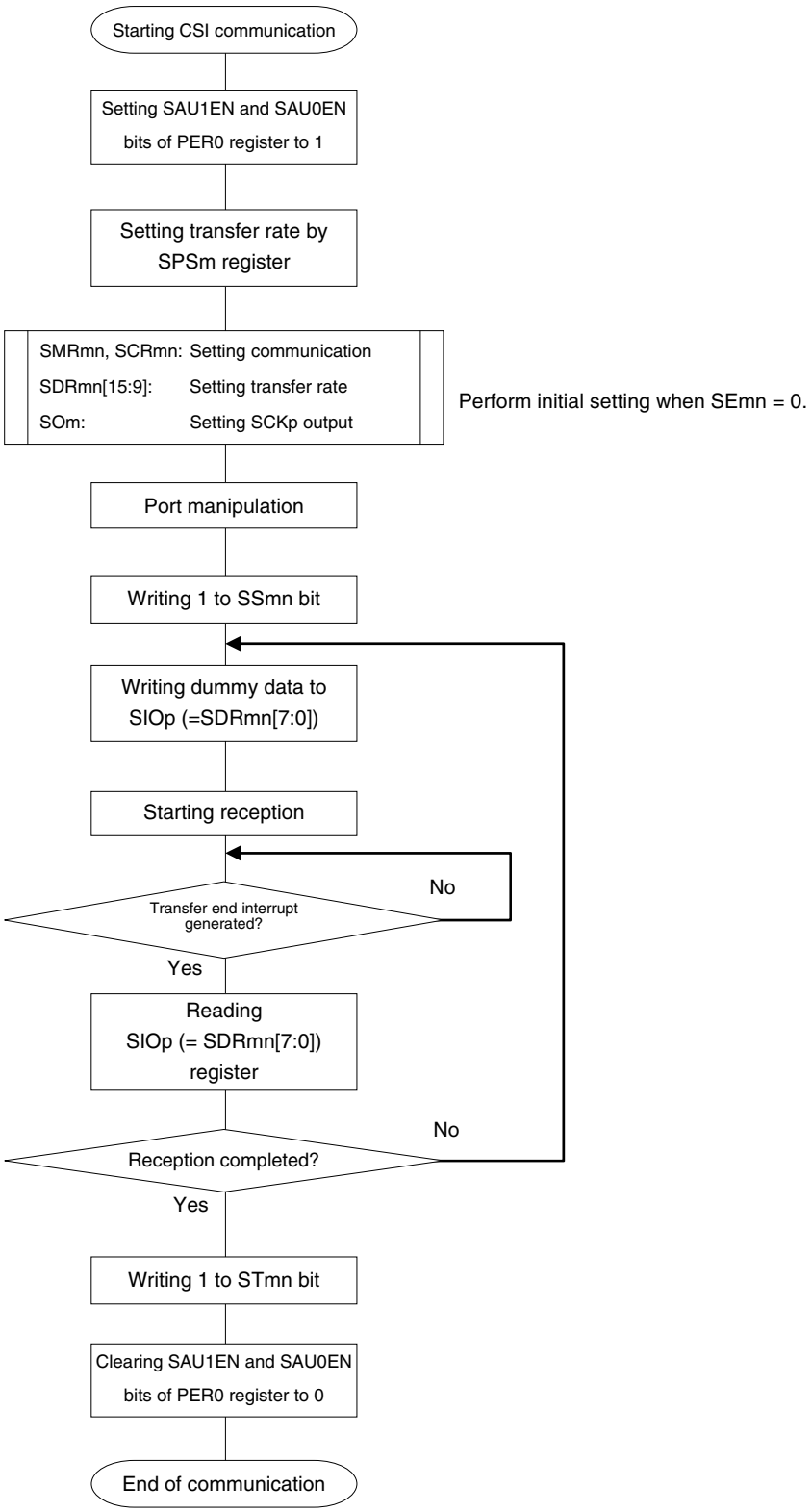
(3) Processing flow (in single-reception mode)

Figure 11-35. 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 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)

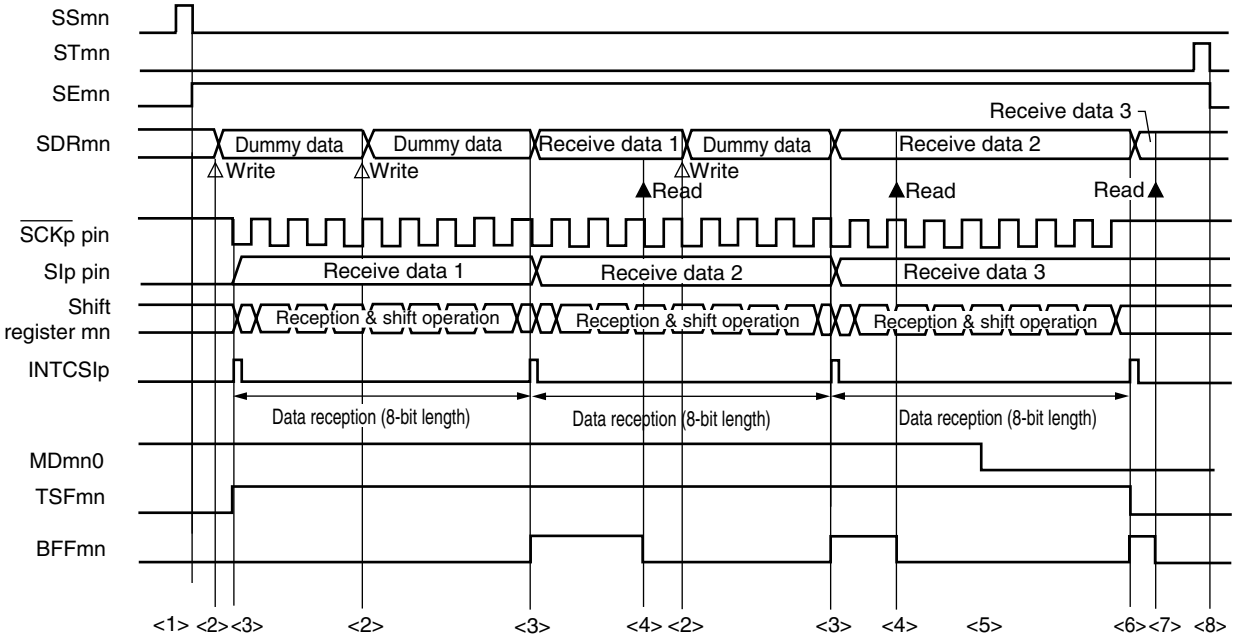
Figure 11-36. Flowchart of Master Reception (in Single-Reception Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous reception mode)

Figure 13-37. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

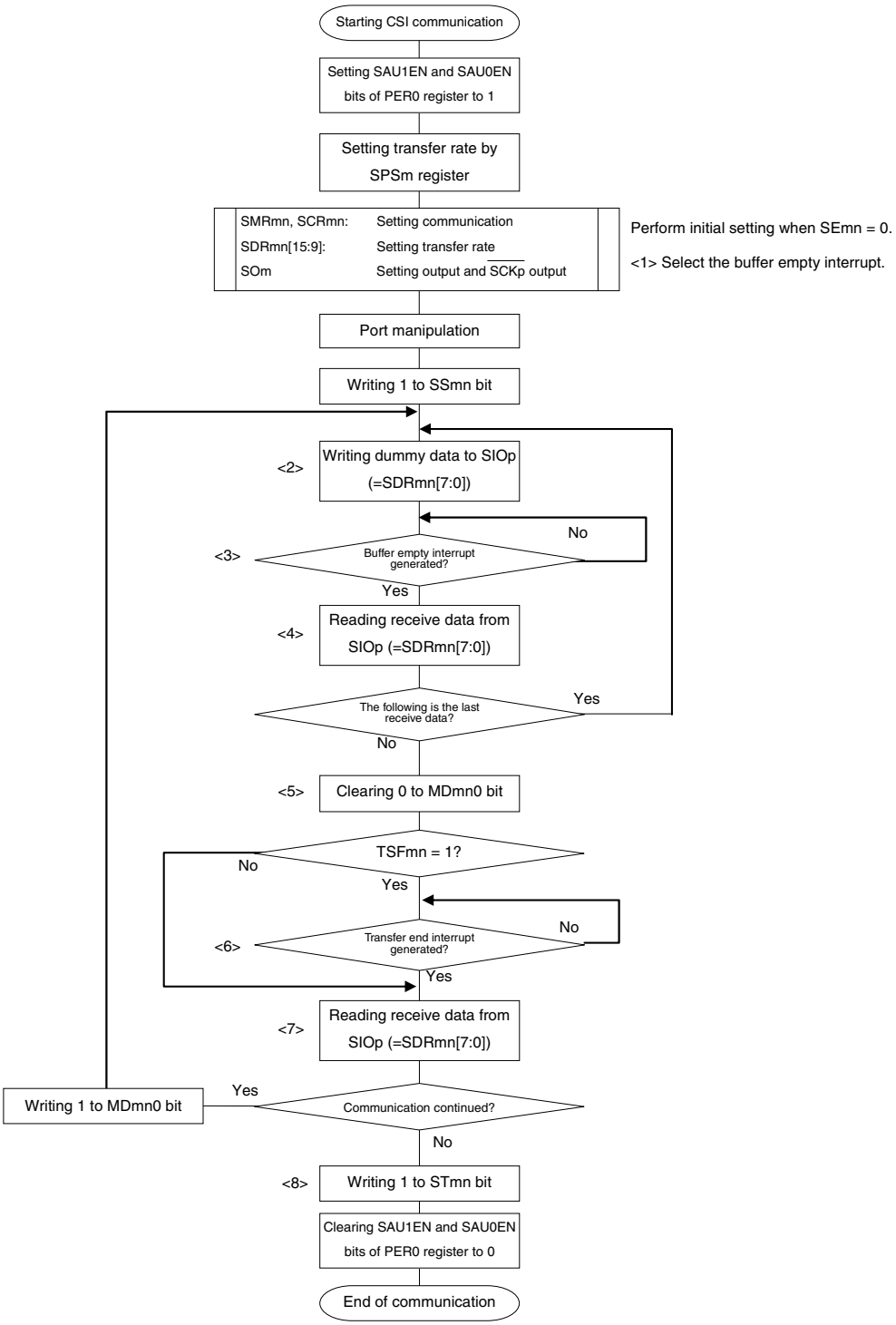


Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-38 Flowchart of Master Reception (in Continuous Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2)
mn = 00 to 02, 10, p: CSI number (p = 00, 01, 10, 11, 20)

Figure 13-38. Flowchart of Master Reception (in Continuous Reception Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-37 Timing Chart of Master Reception (in Continuous Reception Mode)**.

11.5.3 Master transmission/reception

Master transmission/reception is an operation in which the 78K0R/Kx3-C outputs a transfer clock and transmits/receives data to/from another device.

3-Wire Serial I/O	CSI00	CSI01	CSI10	CSI20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	$\overline{\text{SCK00}}$, SI00, SO00	$\overline{\text{SCK01}}$, SI01, SO01	$\overline{\text{SCK10}}$, SI10, SO10	$\overline{\text{SCK20}}$, SI20, SO20
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	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]Note f_{CLK}: System clock frequency			
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse			
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**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10

(1) Register setting

Figure 11-39. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 Note 0/1	CKOm1 Note 0/1	CKOm0 0/1	0	0	0	0	1	SOm2 Note 0/1	SOm1 Note 0/1	SOm0 0/1

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.

0: Serial data output value is "0"
1: Serial data output value is "1"

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 Note 0/1	SOEm1 Note 0/1	SOEm0 0/1

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 Note 0/1	SSm1 Note 0/1	SSm0 0/1

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISm0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

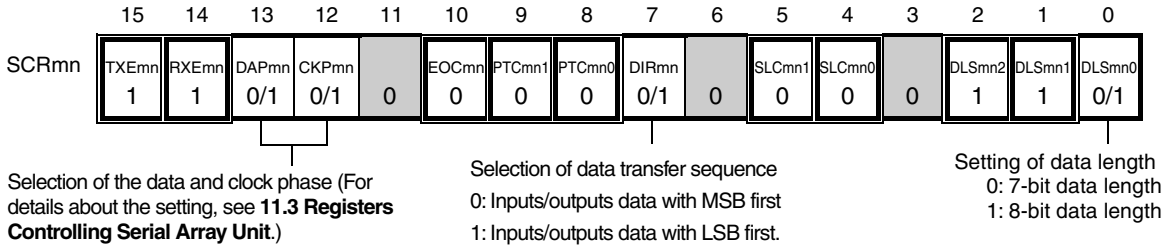
Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

Note Serial array unit 0 only.

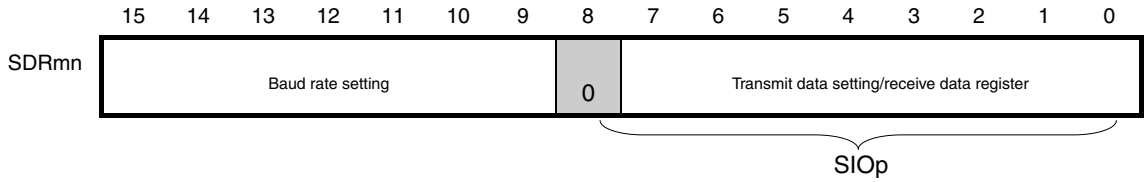
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)
□: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-39. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) (2/2)

(e) Serial communication operation setting register mn (SCRmn)



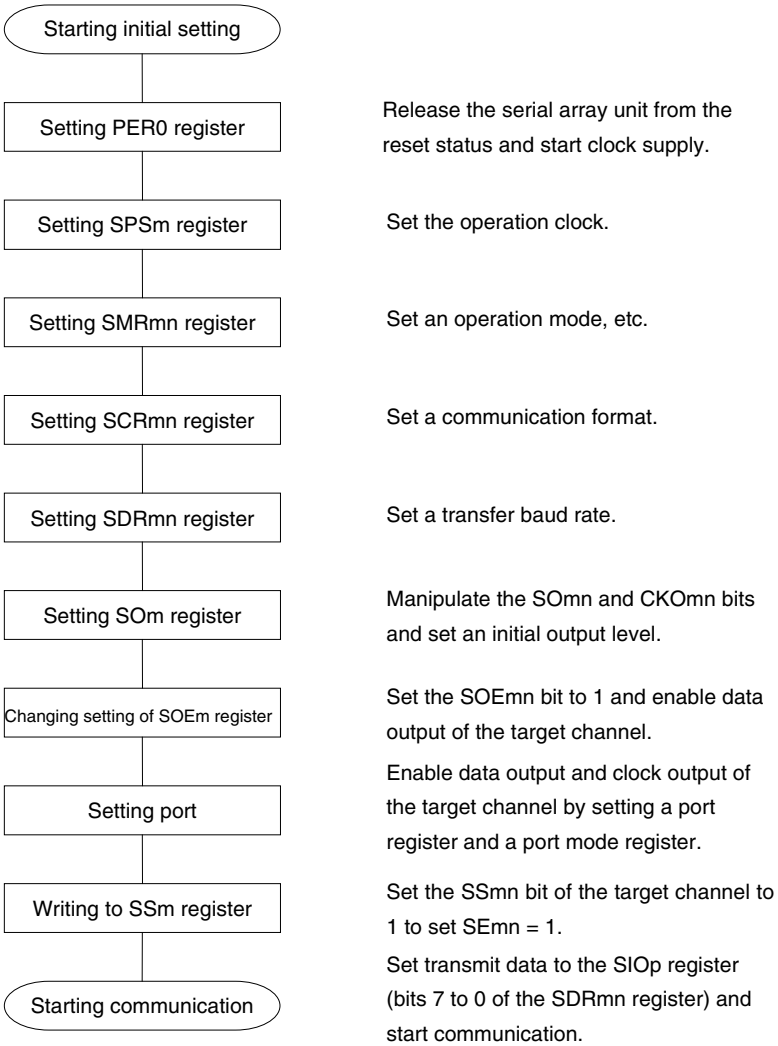
(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)
□: Setting is fixed in the CSI master transmission/reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

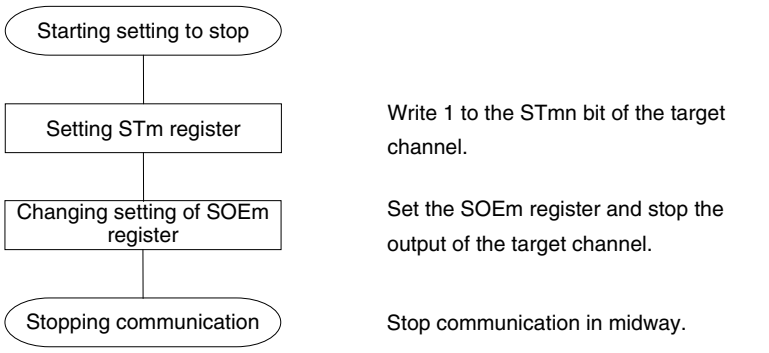
(2) Operation procedure

Figure 11-40. Initial Setting Procedure for Master Transmission/Reception



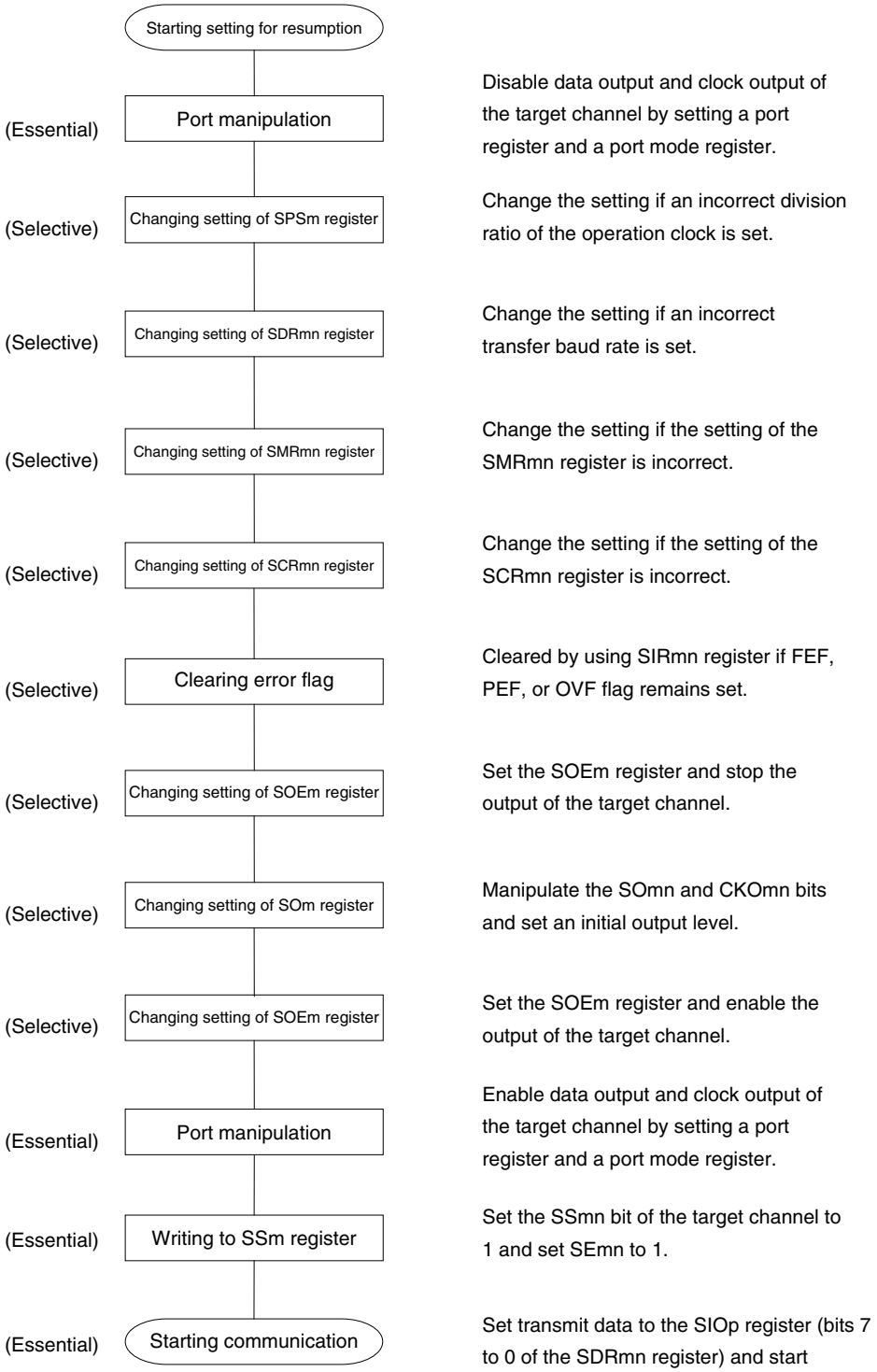
Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 11-41. Procedure for Stopping Master Transmission/Reception



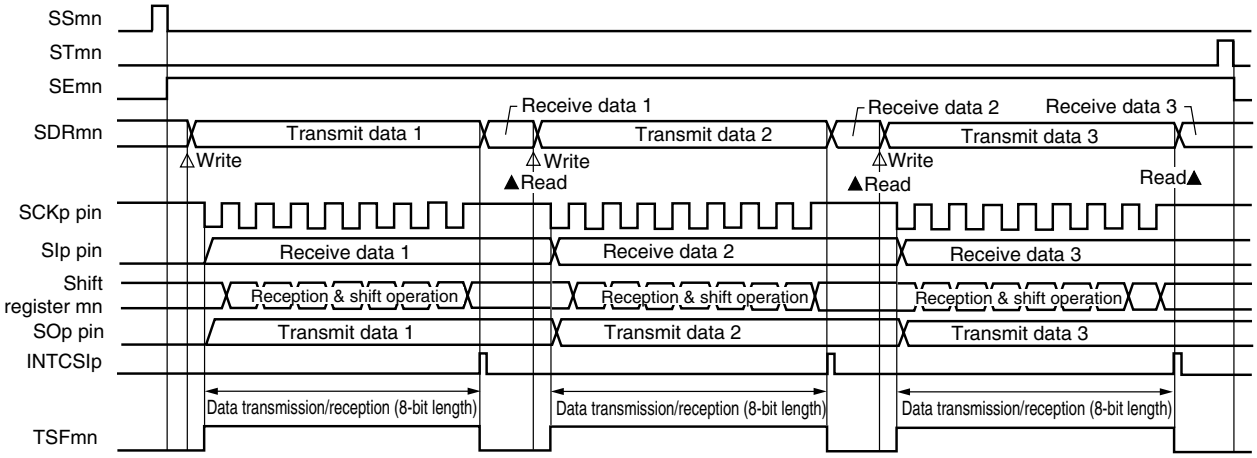
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-42 Procedure for Resuming Master Transmission/Reception**).

Figure 11-42. Procedure for Resuming Master Transmission/Reception



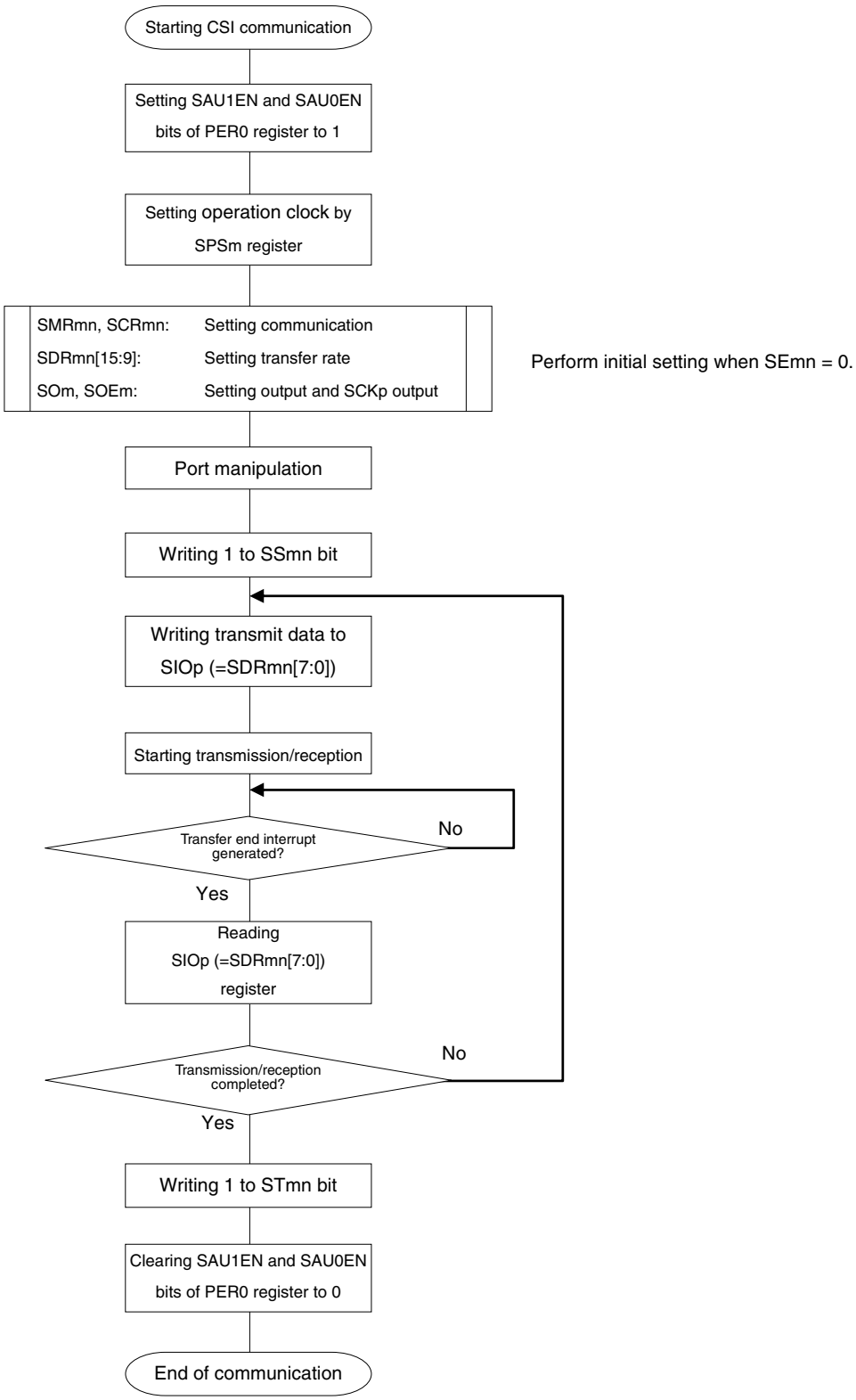
(3) Processing flow (in single-transmission/reception mode)

Figure 11-43. 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 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)

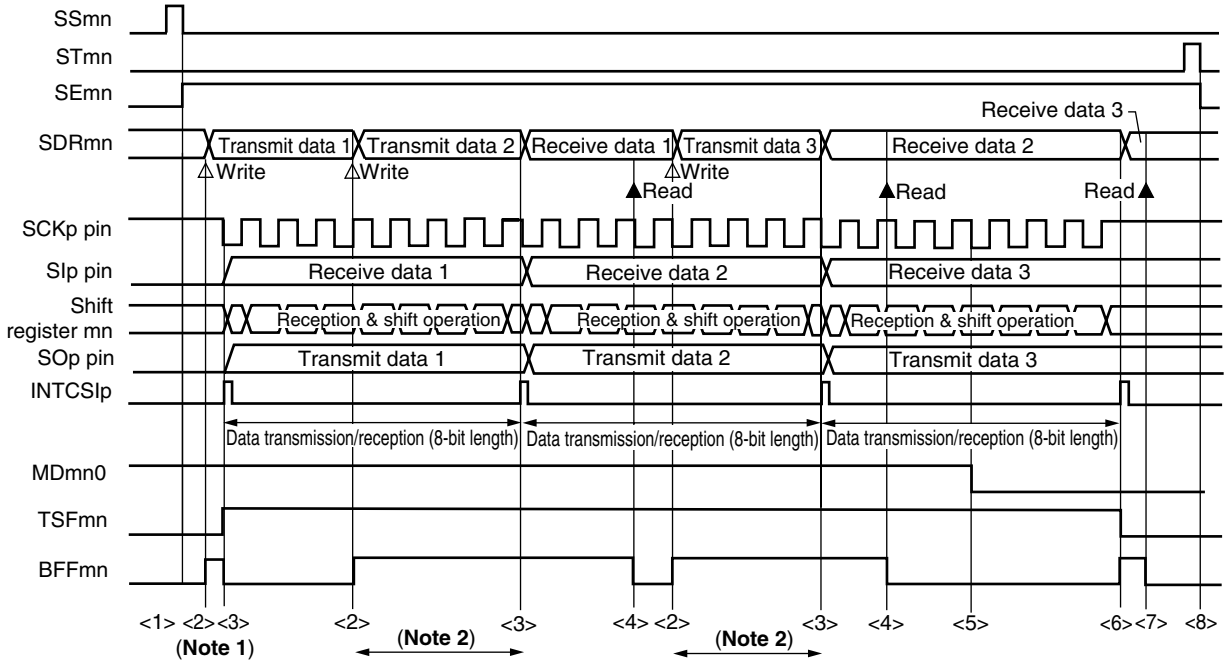
Figure 11-44. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-45. 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.

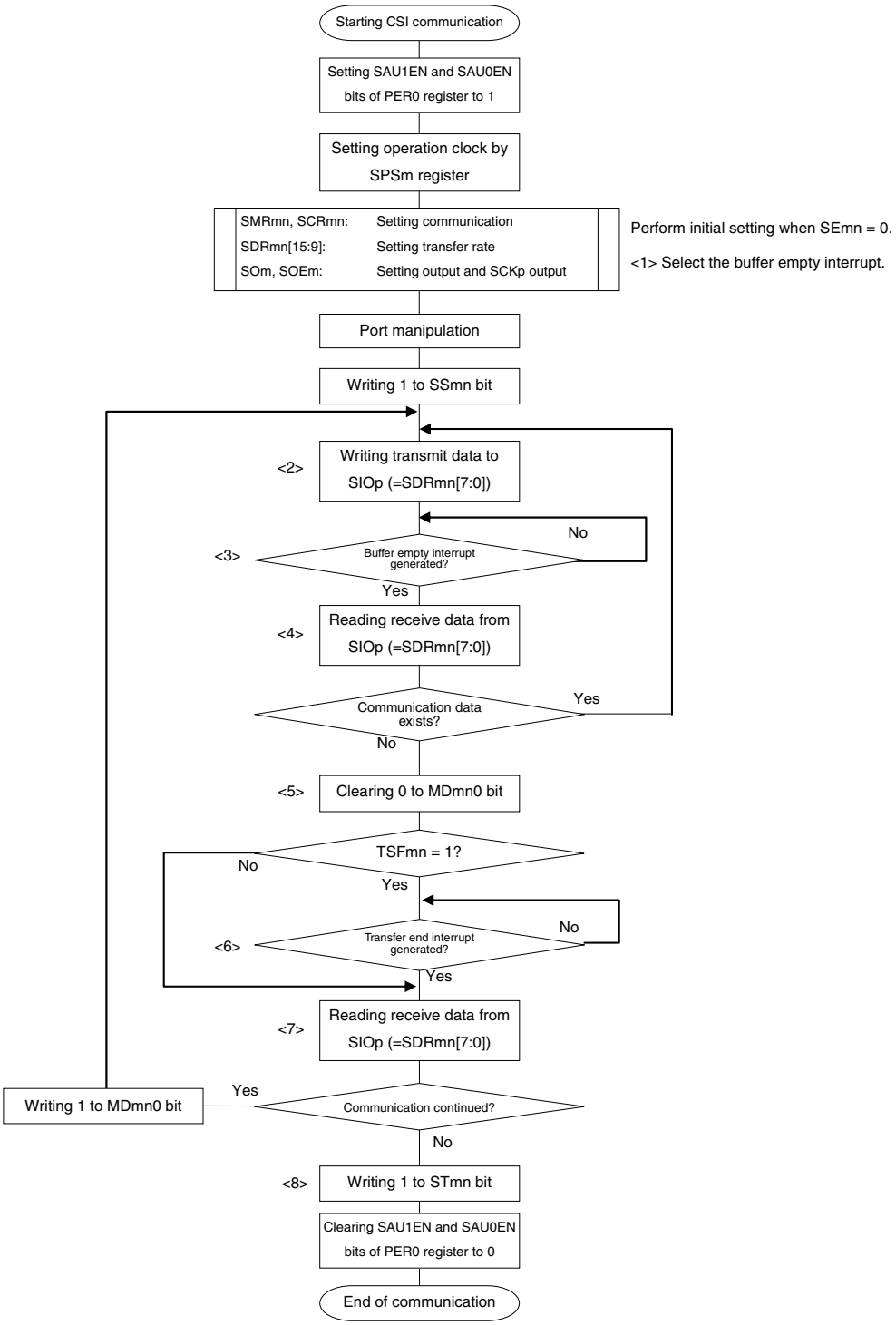
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 11-46 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)

Figure 11-46. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-45 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

11.5.4 Slave transmission

Slave transmission is an operation in which the 78K0R/Kx3-C 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	CSI20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	$\overline{SCK00}$, SO00	$\overline{SCK01}$, SO01	$\overline{SCK10}$, SO10	$\overline{SCK20}$, SO20
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
Transfer rate	Max. f_{MCK}/6 [Hz] Notes 1, 2			
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse			
Data direction	MSB or LSB first			

Notes 1. Because the external serial clock input to pins $\overline{SCK00}$, $\overline{SCK01}$, $\overline{SCK10}$, and $\overline{SCK20}$ is sampled internally and used, the maximum transfer rate is f_{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**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10

(1) Register setting

Figure 11-47. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	1	SOm2 Note 0/1	SOm1 Note 0/1	SOm0 0/1

0: Serial data output value is "0"
1: Serial data output value is "1"

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 Note 0/1	SOEm1 Note 0/1	SOEm0 0/1

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 Note 0/1	SSm1 Note 0/1	SSm0 0/1

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISm0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

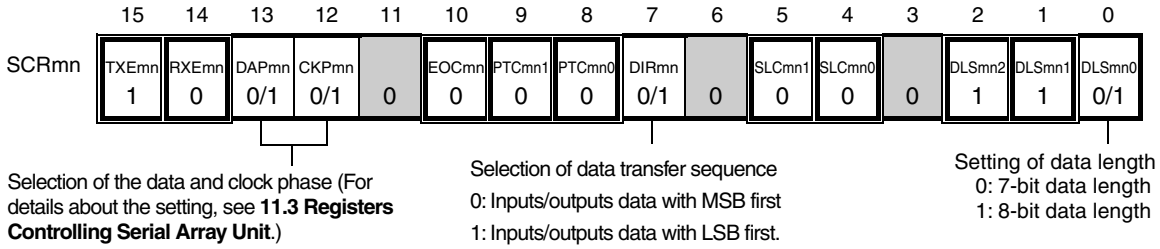
Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

Note Serial array unit 0 only.

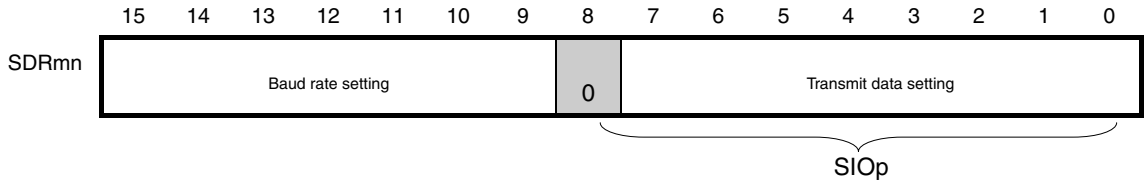
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)
□: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-47. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) (2/2)

(e) Serial communication operation setting register mn (SCRmn)



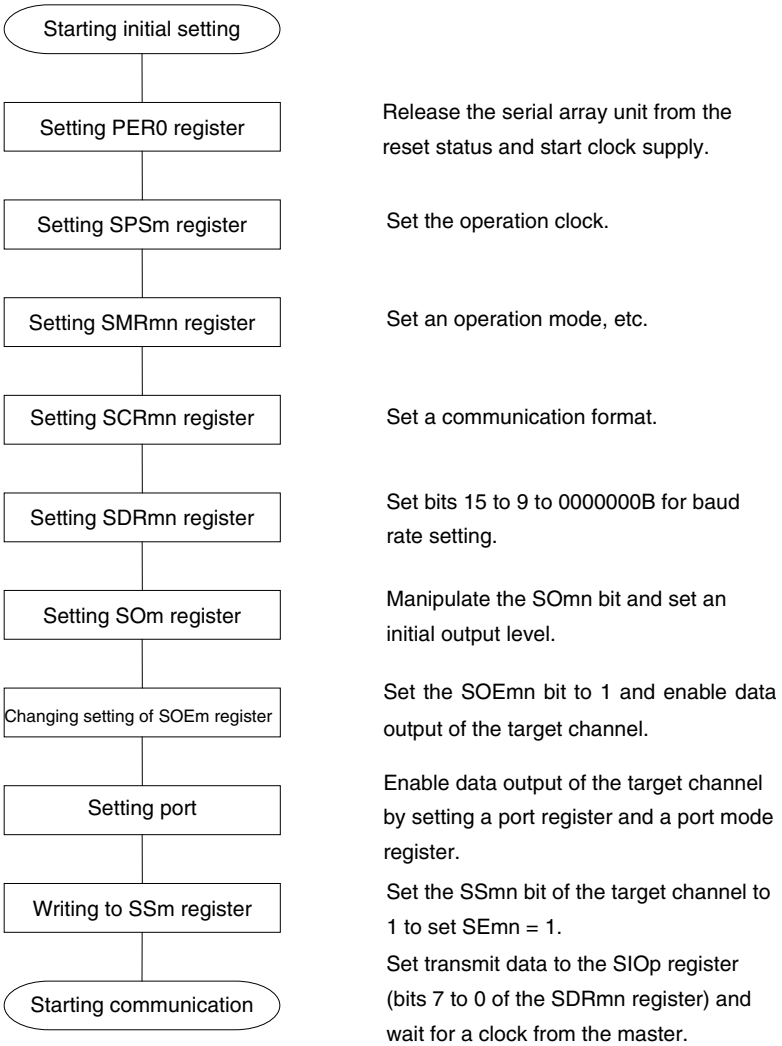
(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)
□: Setting is fixed in the CSI slave transmission mode, ■ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

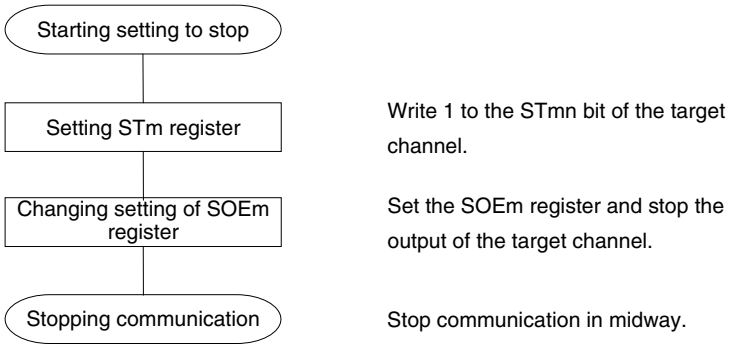
(2) Operation procedure

Figure 11-48 Initial Setting Procedure for Slave Transmission



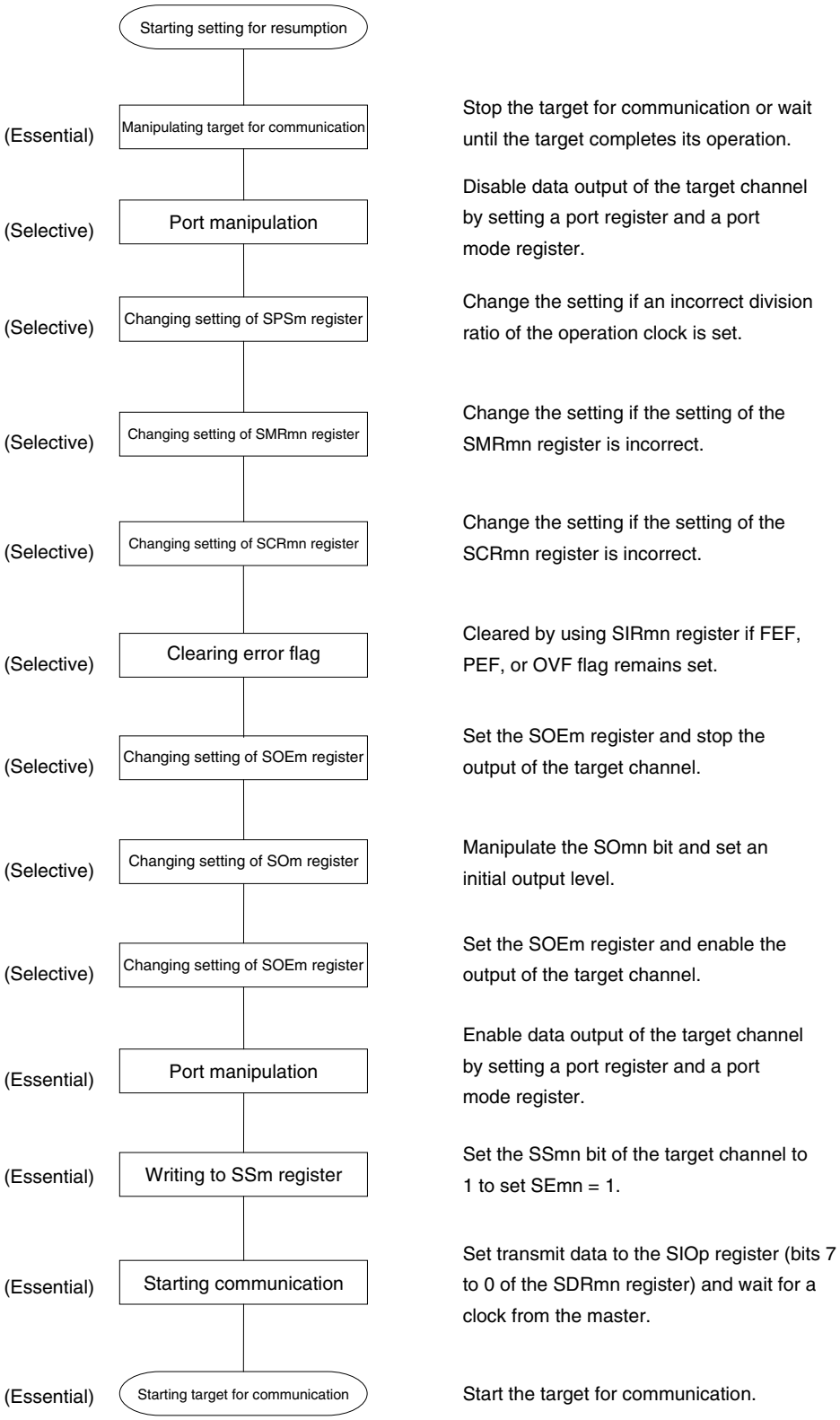
Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 11-49 Procedure for Stopping Slave Transmission



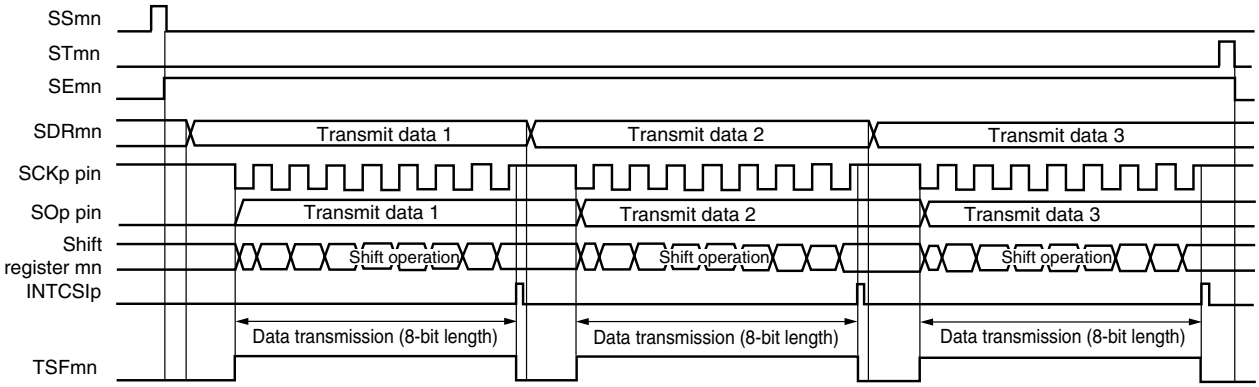
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-50 Procedure for Resuming Slave Transmission**).

Figure 11-50 Procedure for Resuming Slave Transmission



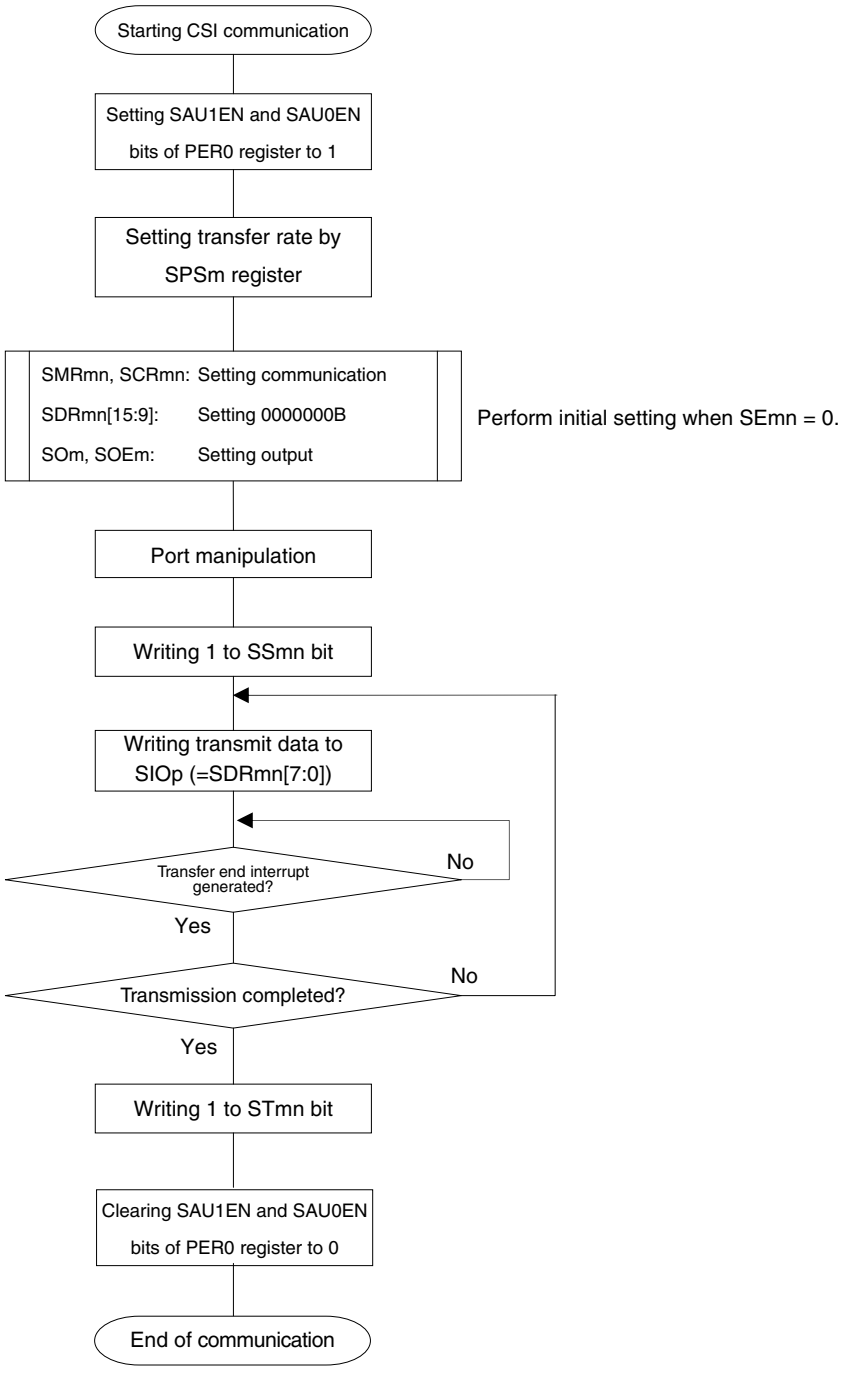
(3) Processing flow (in single-transmission mode)

Figure 11-51. 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 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)

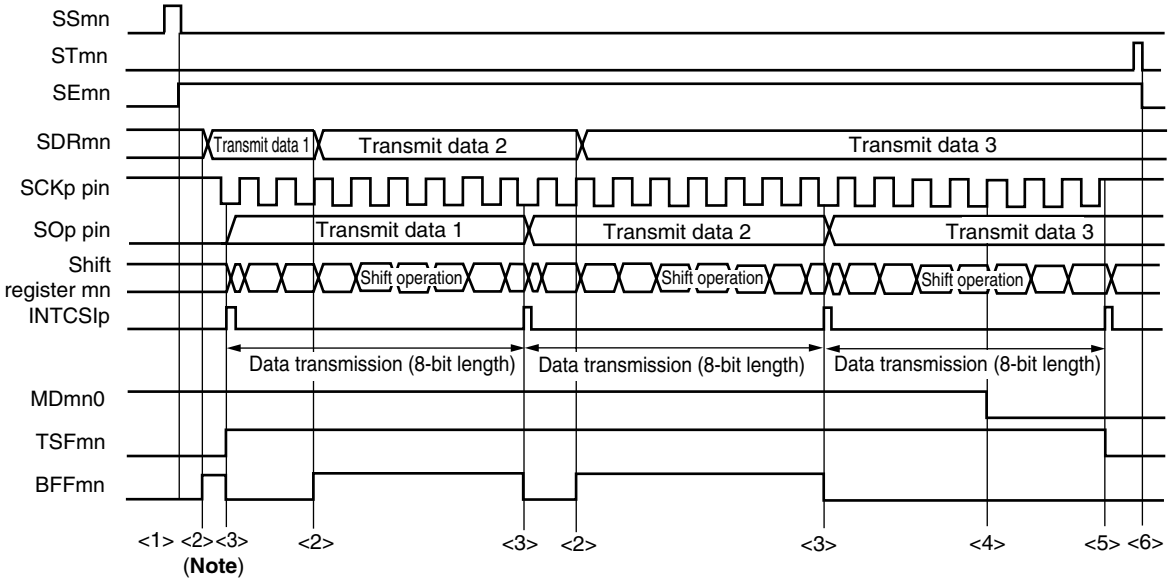
Figure 11-52. Flowchart of Slave Transmission (in Single-Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

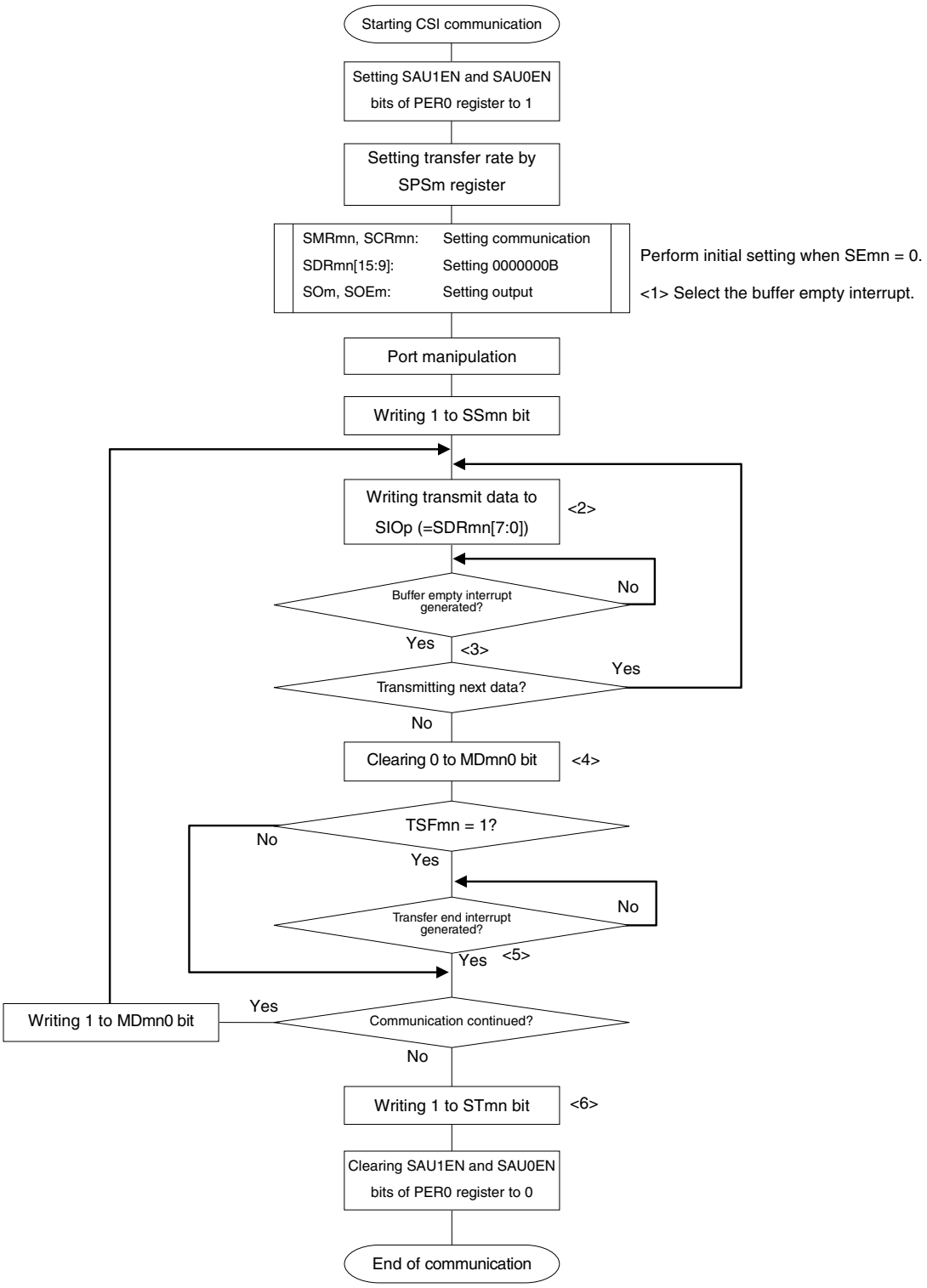
Figure 11-53. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



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 11-54. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-53 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

11.5.5 Slave reception

Slave reception is an operation in which the 78K0R/Kx3-C 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	CSI20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	$\overline{SCK00}$, SI00	$\overline{SCK01}$, SI01	$\overline{SCK10}$, SI10	$\overline{SCK20}$, SI20
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
Transfer rate	Max. f_{MCK}/6 [Hz] Notes 1, 2.			
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse			
Data direction	MSB or LSB first			

Notes 1. Because the external serial clock input to pins $\overline{SCK00}$, $\overline{SCK01}$, $\overline{SCK10}$, and $\overline{SCK20}$ is sampled internally and used, the maximum transfer rate is f_{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**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10

(1) Register setting

Figure 11-55. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) (1/2)

(a) Serial output register m (SOm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	1	SOm2 ×	SOm1 ×	SOm0 ×

(b) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 ×	SOEm0 ×

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 Note 0/1	SSm1 Note 0/1	SSm0 0/1

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISm0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

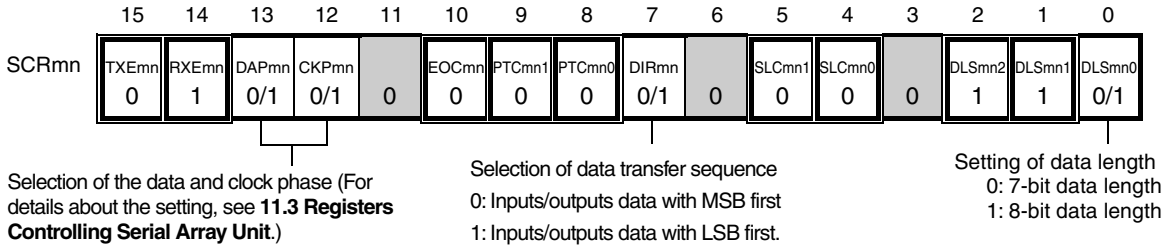
Interrupt sources of channel n
0: Transfer end interrupt

Note Serial array unit 0 only.

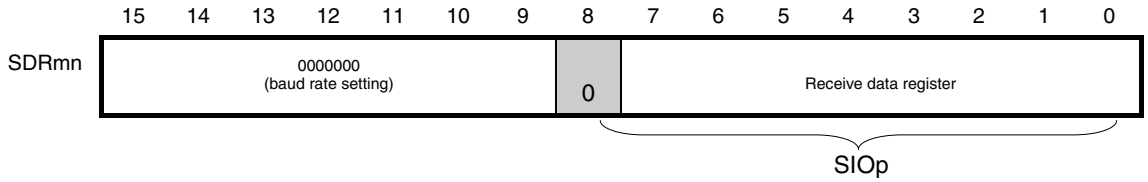
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)
□: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-55. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) (2/2)

(e) Serial communication operation setting register mn (SCRmn)



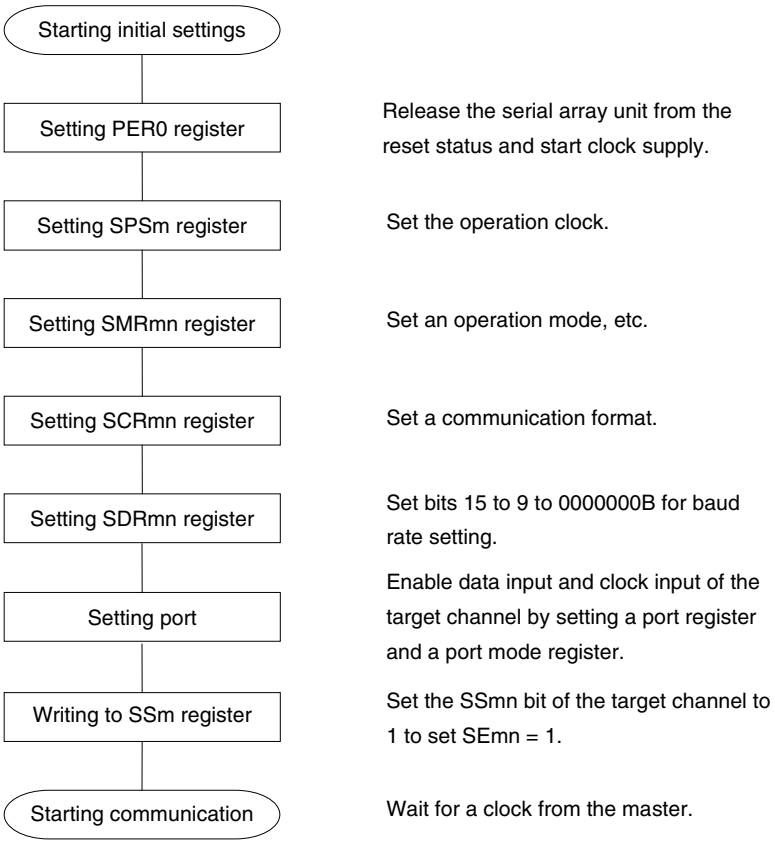
(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)
□: Setting is fixed in the CSI slave reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-56. Initial Setting Procedure for Slave Reception



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 11-57. Procedure for Stopping Slave Reception

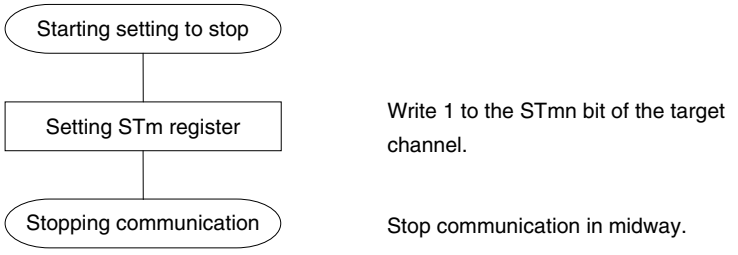
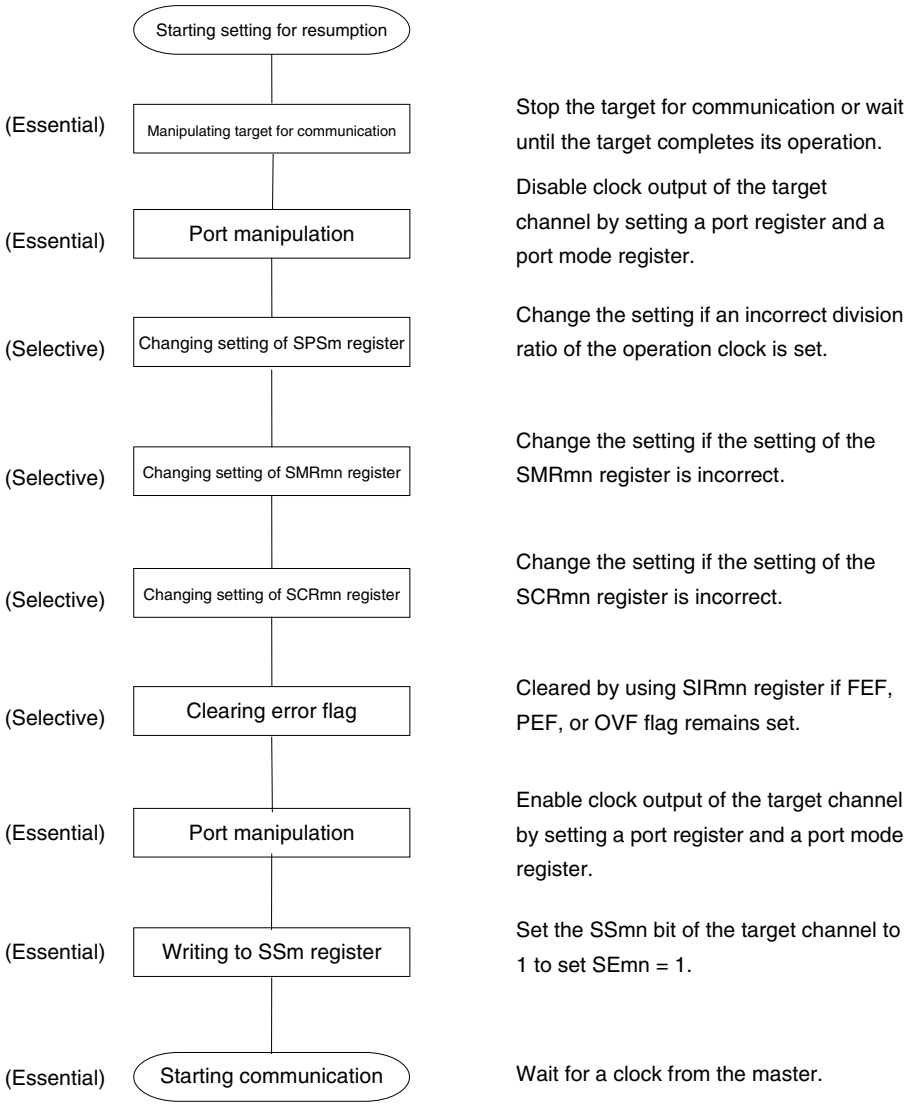
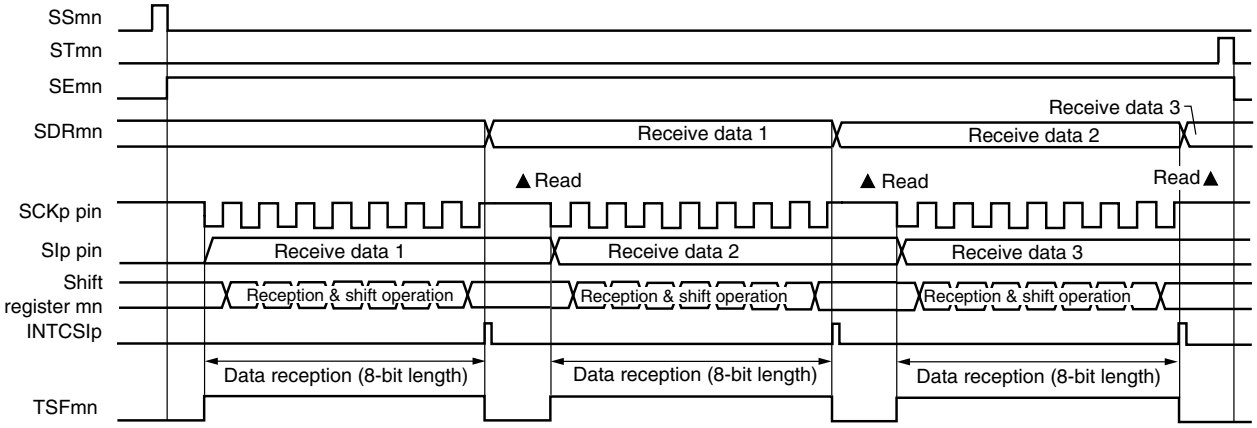


Figure 11-58. Procedure for Resuming Slave Reception



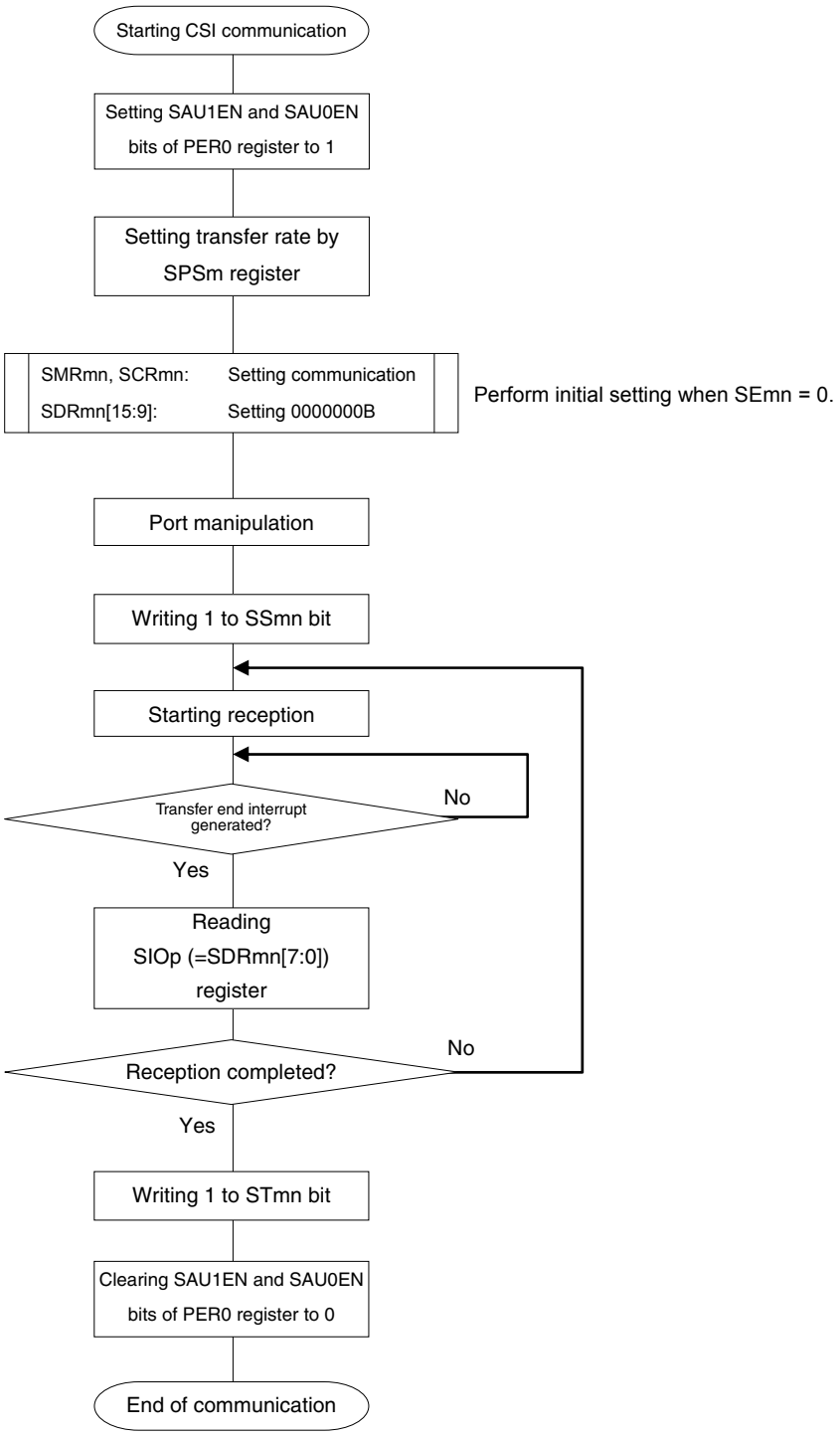
(3) Processing flow (in single-reception mode)

Figure 11-59. 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 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)

Figure 11-60. Flowchart of Slave Reception (in Single-Reception Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

11.5.6 Slave transmission/reception

Slave transmission/reception is an operation in which the 78K0R/Kx3-C 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	CSI20
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	$\overline{SCK00}$, SI00, SO00	$\overline{SCK01}$, SI01, SO01	$\overline{SCK10}$, SI10, SO10	$\overline{SCK20}$, SI20, SO20
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 or 8 bits			
Transfer rate	Max. f_{MCK}/6 [Hz] Notes 1, 2.			
Data phase	Selectable by DAPmn bit • DAPmn = 0: Data I/O starts from the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.			
Clock phase	Selectable by CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse			
Data direction	MSB or LSB first			

Notes 1. Because the external serial clock input to pins $\overline{SCK00}$, $\overline{SCK01}$, $\overline{SCK10}$, and $\overline{SCK20}$ is sampled internally and used, the maximum transfer rate is f_{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**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10

(1) Register setting

Figure 11-61. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	1	SOm2 Note 0/1	SOm1 Note 0/1	SOm0 0/1

0: Serial data output value is "0"
1: Serial data output value is "1"

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 Note 0/1	SOEm1 Note 0/1	SOEm0 0/1

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 Note 0/1	SSm1 Note 0/1	SSm0 0/1

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISm0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt sources of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

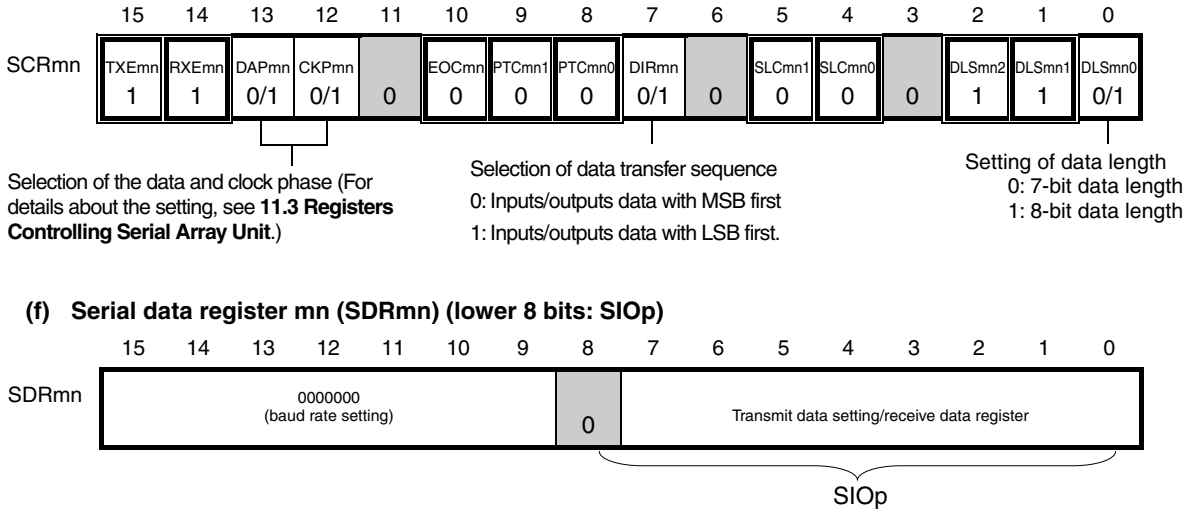
Note Serial array unit 0 only.

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 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)
□: Setting is fixed in the CSI master transmission mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-61. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI20) (2/2)

(e) Serial communication operation setting register mn (SCRmn)



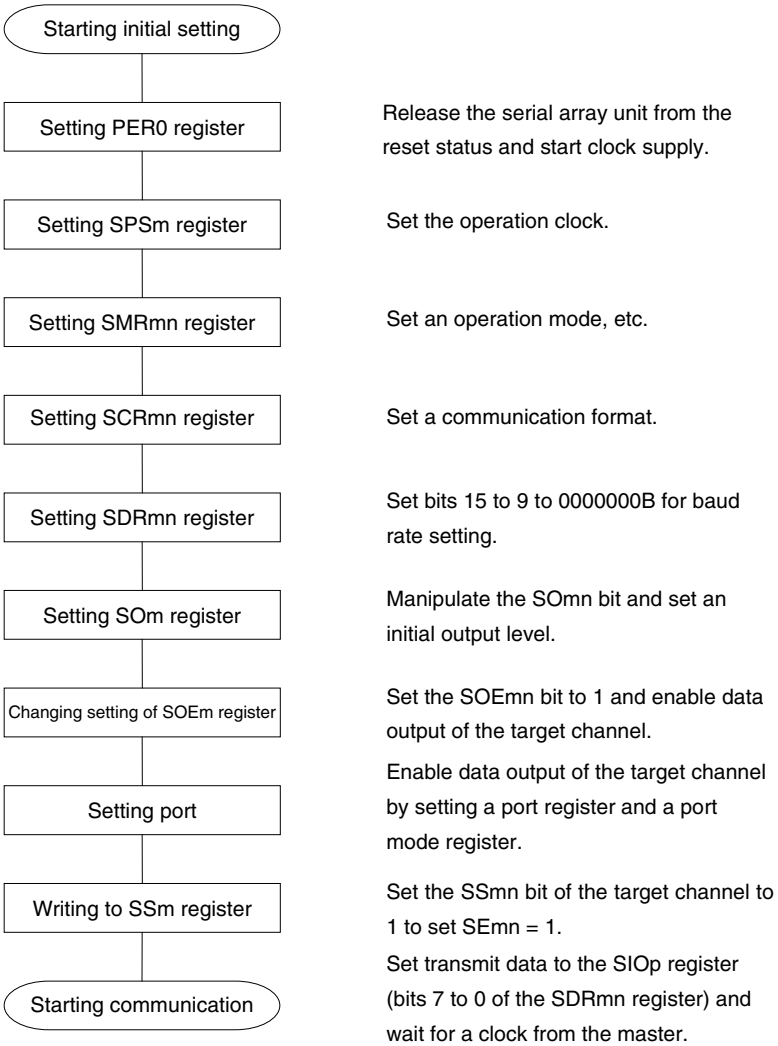
(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

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 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)
□ : 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

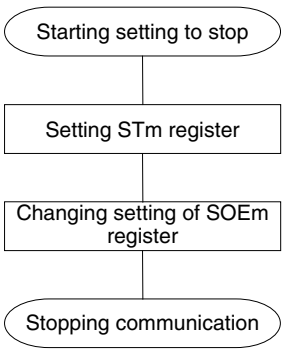
Figure 11-62. Initial Setting Procedure for Slave Transmission/Reception



Cautions 1. After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

2. Be sure to set transmit data to the SIOp register before the clock from the master is started.

Figure 11-63. Procedure for Stopping Slave Transmission/Reception



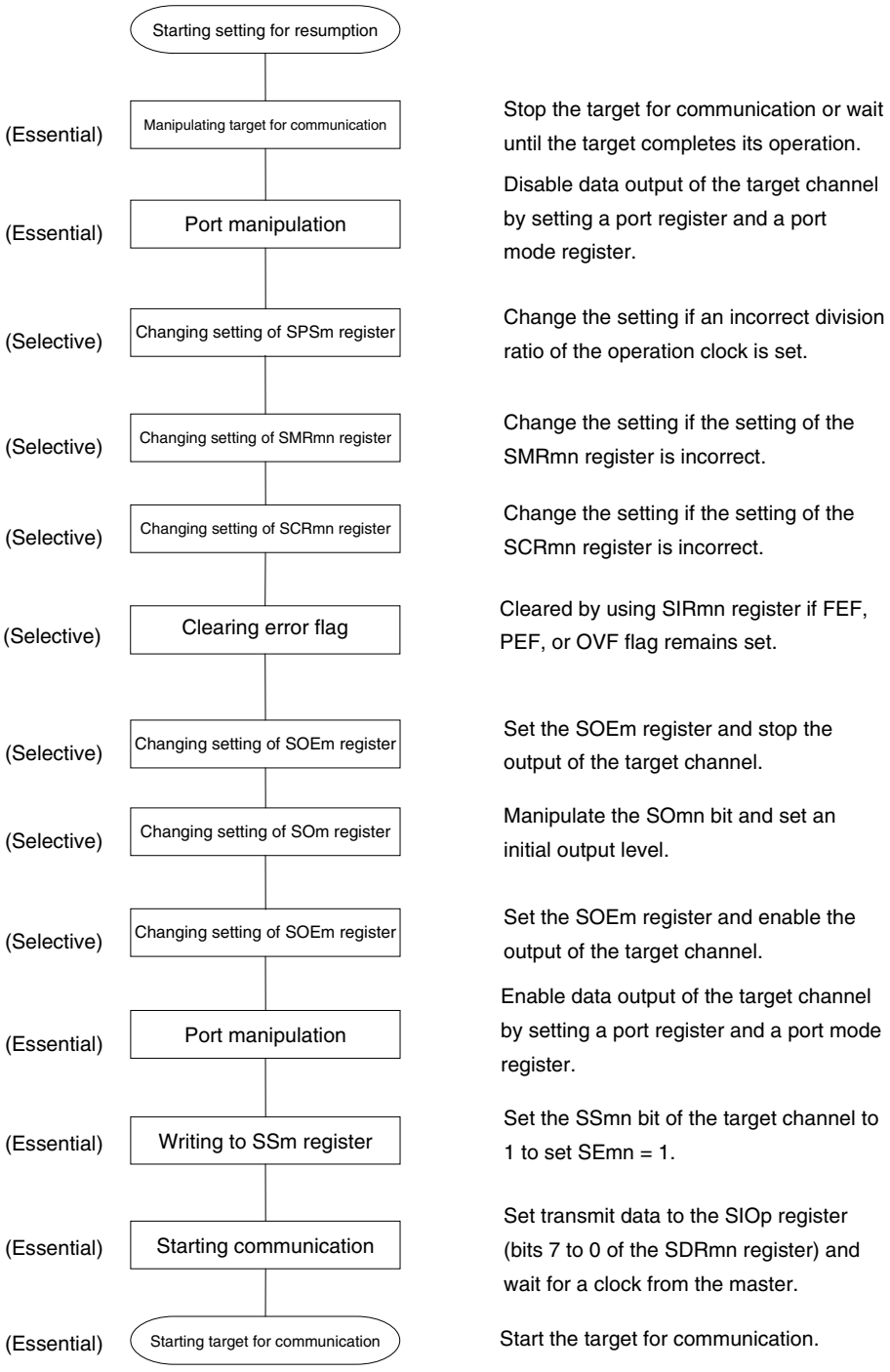
Write 1 to the STmn bit of the target channel.

Set the SOEm register and stop the output of the target channel.

Stop communication in midway.

Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-64 Procedure for Resuming Slave Transmission/Reception**).

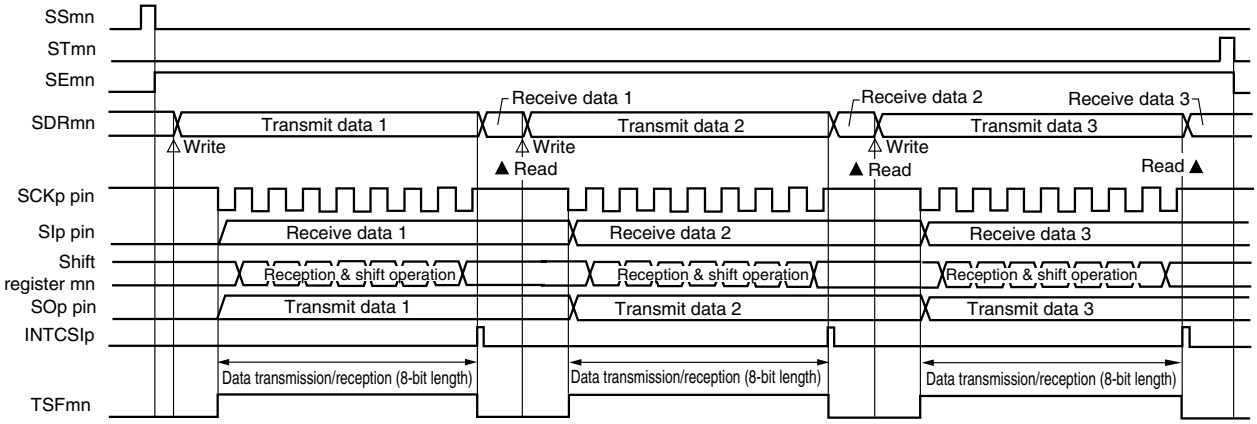
Figure 11-64. Procedure for Resuming Slave Transmission/Reception



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

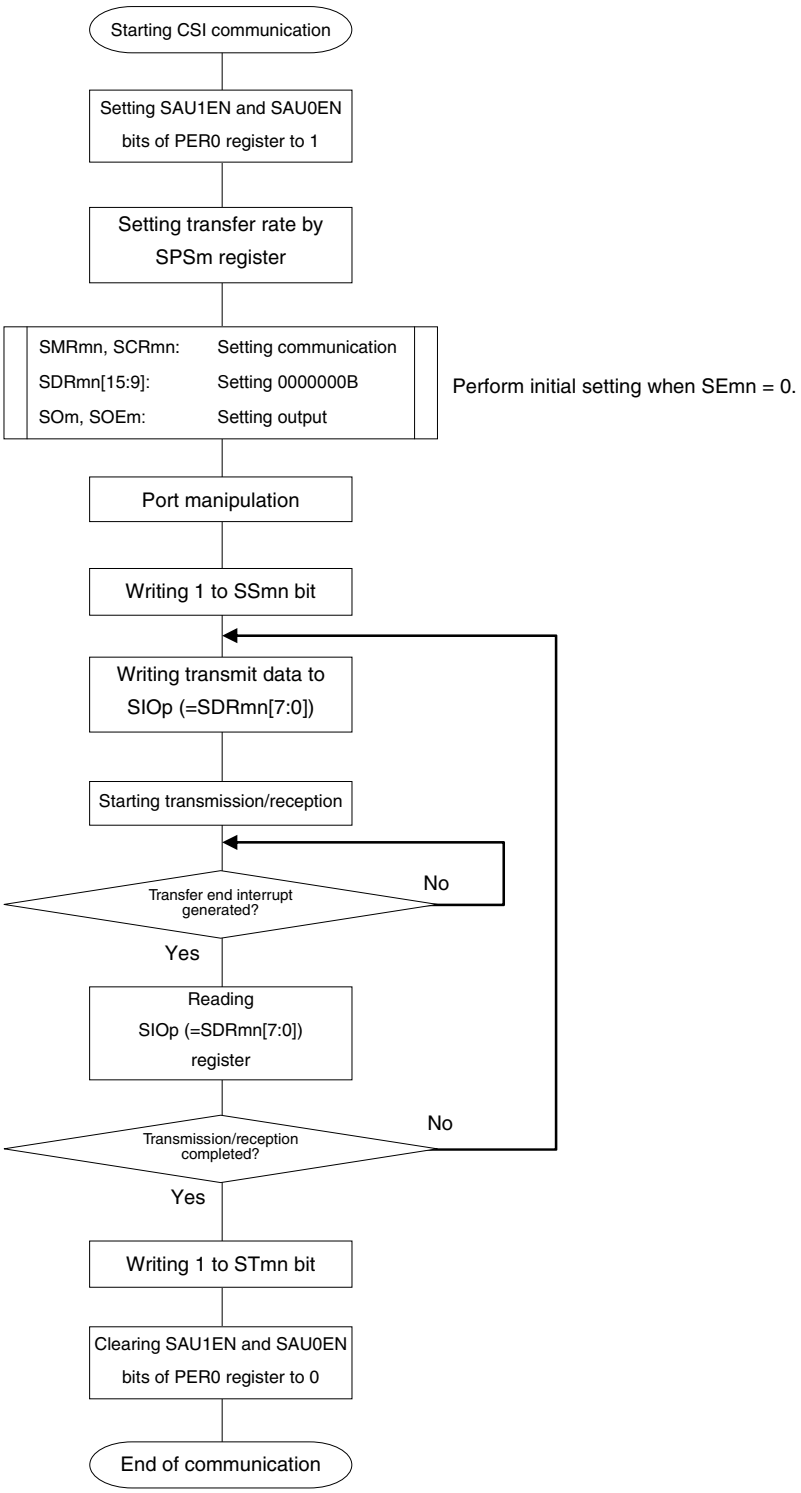
(3) Processing flow (in single-transmission/reception mode)

Figure 11-65. 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 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)

Figure 11-66. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)

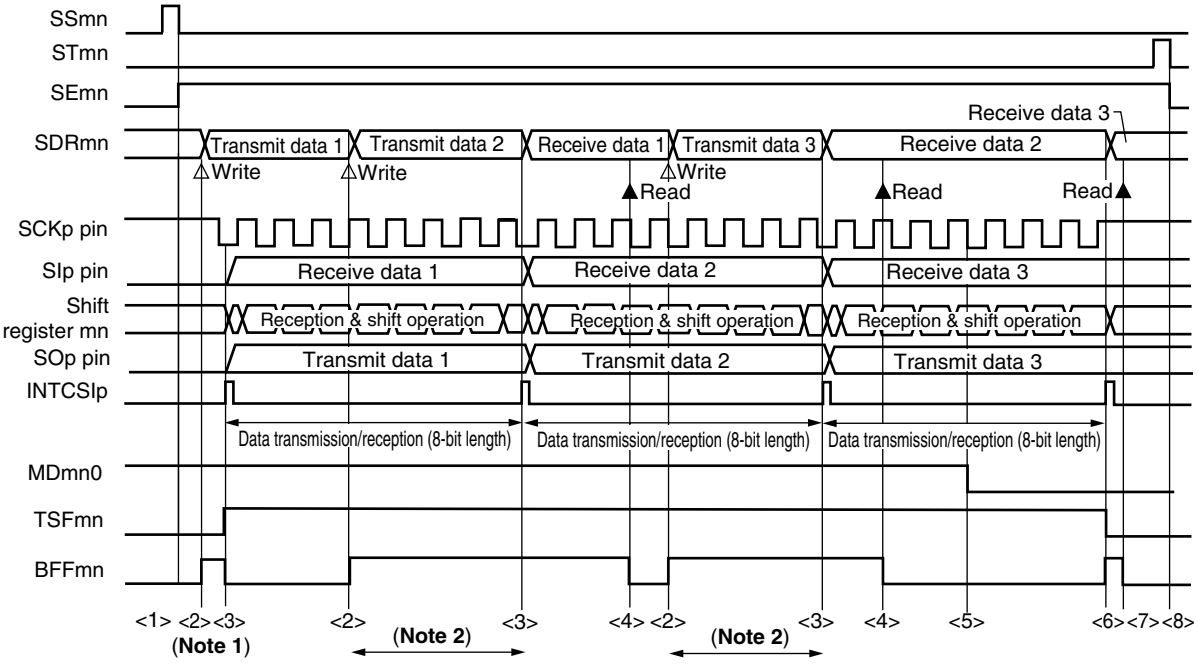


Cautions 1. After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

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 11-67. 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.

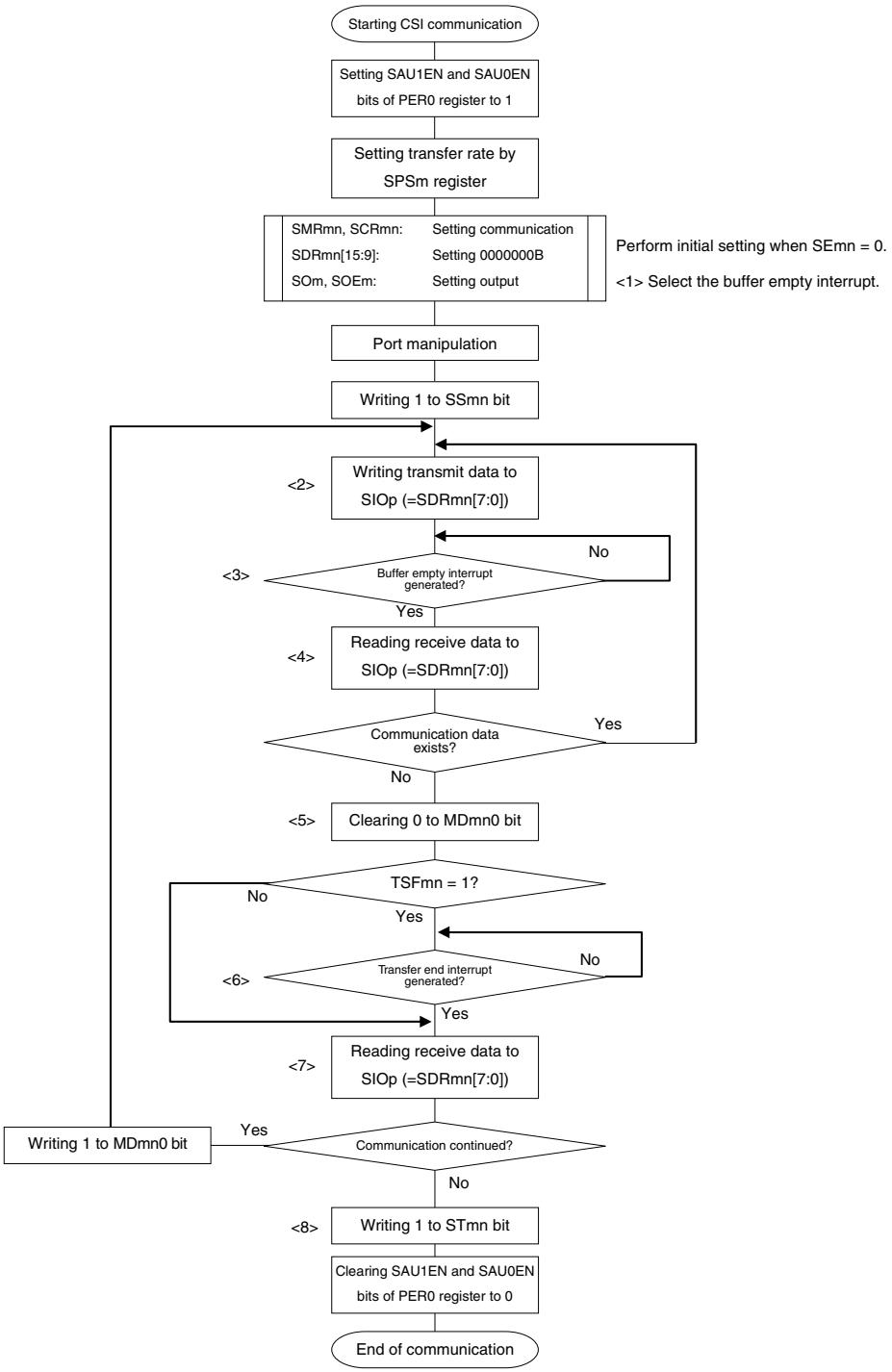
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 11-68 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10,
p: CSI number (p = 00, 01, 10, 20)

Figure 11-68. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Cautions 1. After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} 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 11-67 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

11.5.7 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI00, CSI01, CSI10, CSI20) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (f_{SCK}) supplied by master}[Note] [Hz]

Note The permissible maximum transfer clock frequency is f_{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 2), mn = 00 to 02, 10

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11-2. Selection of Operation Clock

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note 1	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	313 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0 Note 2, Setting prohibited if m = 1	
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	313 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0 Note 2, Setting prohibited if m = 1	
Other than above									Setting prohibited	

Notes 1. When changing the clock selected for f_{CLK} (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 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAU0 can be used at the fixed division ratio of the subsystem clock regardless of the f_{CLK} frequency by setting TAU0 and SAU0 as follows.

<TAU0> Select $f_{SUB}/4$ as the input clock of channel 2 of TAU0. (Set TIS02 to 1.)

<SAU0> Select INTTM02 by using the SPS0 register.

However, when changing f_{CLK}, SAU0 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 2), mn = 00 to 02, 10

11.5.8 Procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI20) communication

The procedure for processing errors that occurred during 3-wire serial I/O (CSI00, CSI01, CSI10, CSI20) communication is described in Figure 11-69.

Figure 11-69. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2), mn = 00 to 02, 10

11.6 Operation of UART (UART0, UART1, UART2) 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

UART0 uses channels 0 and 1 of SAU0.
UART1 uses channels 2 and 3 of SAU0.
UART2 uses channels 0 and 1 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	–
	1	CSI01		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–

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 two types of communication operations.

- UART transmission (See **11.6.1**.)
- UART reception (See **11.6.2**.)

11.6.1 UART transmission

UART transmission is an operation to transmit data from the 78K0R/Kx3-C 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
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	TxD0	TxD1	TxD2
Interrupt	INTST0	INTST1	INTST2
	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] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps]Note		
Data phase	Forward output (default: high level) Reverse output (default: low level)		
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity		
Stop bit	The following selectable • Appending 1 bit • Appending 2 bits		
Data direction	MSB or LSB first		

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

(1) Register setting

Figure 11-70. Example of Contents of Registers for UART Transmission of UART (UART0, UART1, UART2) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel to 1.

SOm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	1	SOm2 Note 1 0/1 Note 2	SOm1 Note ×	SOm0 0/1 Note 2

0: Serial data output value is “0”
1: Serial data output value is “1”

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

SOEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 Note 1 0/1	SOEm1 ×	SOEm0 0/1

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

SSm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 Note 1 0/1	SSm1 ×	SSm0 0/1

(d) Serial output level register m (SOLm) … Sets only the bits of the target channel.

SOLm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	SOLm2 Note 1 0/1	0	SOLm0 0/1

0: Forward (normal) transmission
1: Reverse transmission

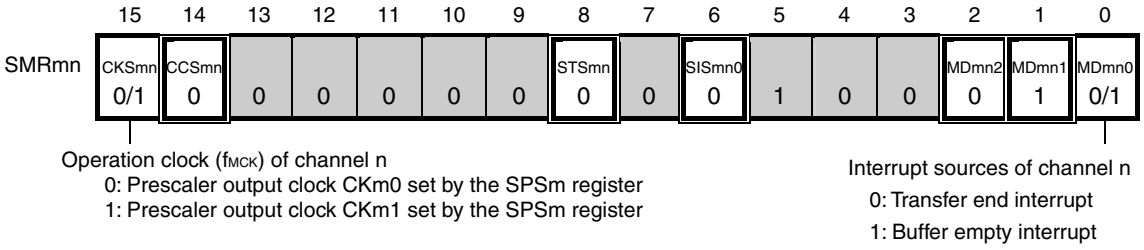
Notes 1. Serial array unit 0 only.

2. 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.

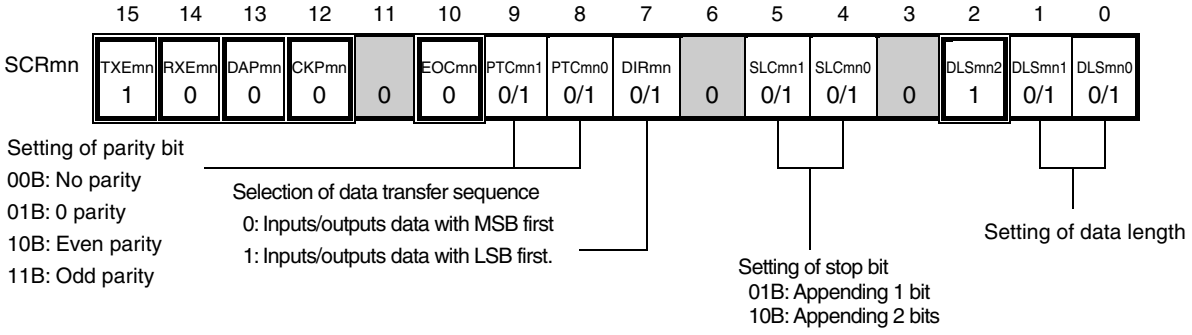
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2) , mn = 00, 02, 10,
q: UART number (q = 0 to 2)
□: Setting is fixed in the UART transmission mode, ■: Setting disabled (fixed by hardware)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-70. Example of Contents of Registers for UART Transmission of UART (UART0, UART1, UART2) (2/2)

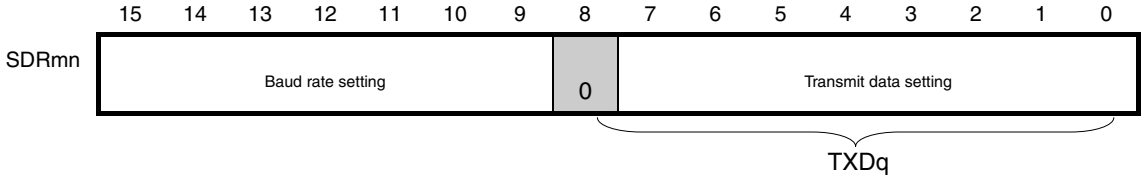
(e) Serial mode register mn (SMRmn)



(f) Serial communication operation setting register mn (SCRmn)



(g) Serial data register mn (SDRmn) (lower 8 bits: TXDq)

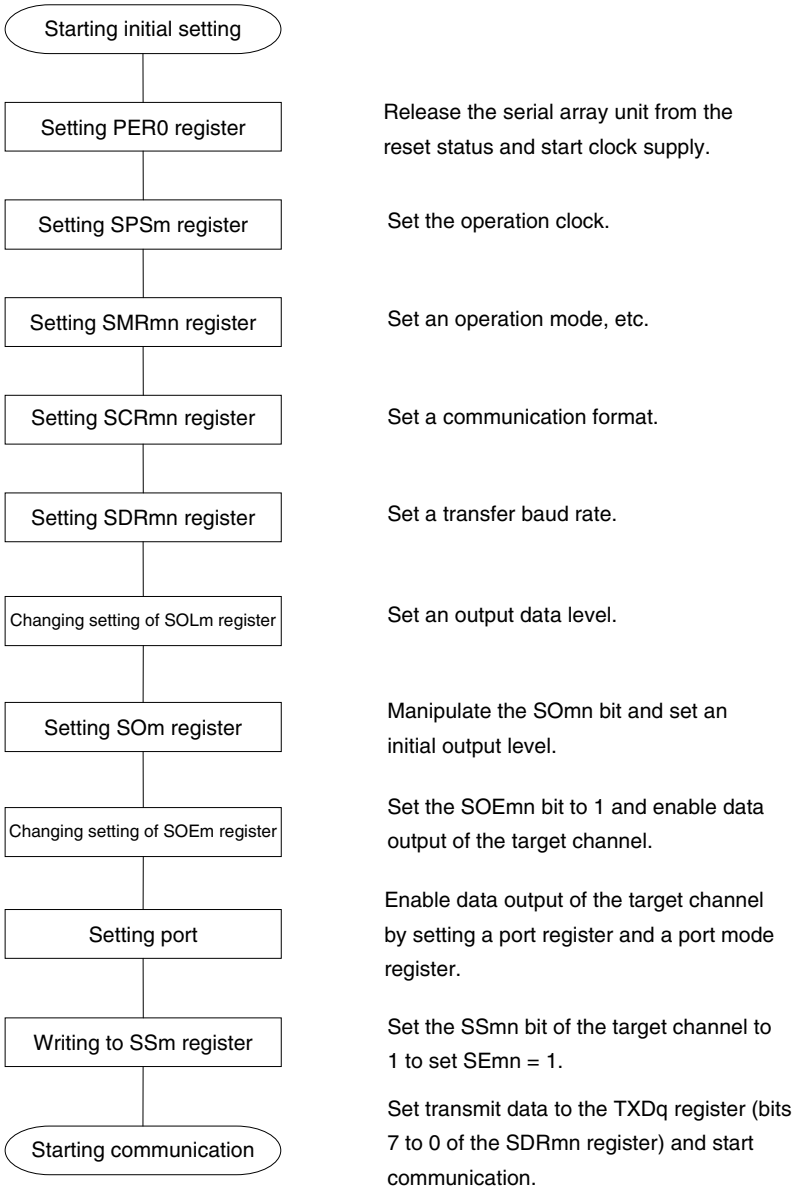


Note Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2)
□: Setting is fixed in the UART transmission mode, ■ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

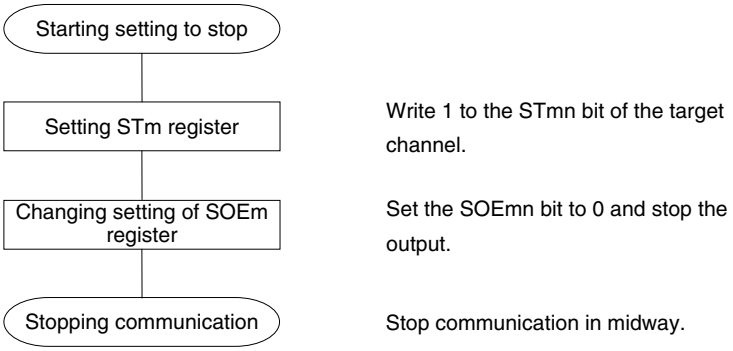
(2) Operation procedure

Figure 11-71. Initial Setting Procedure for UART Transmission



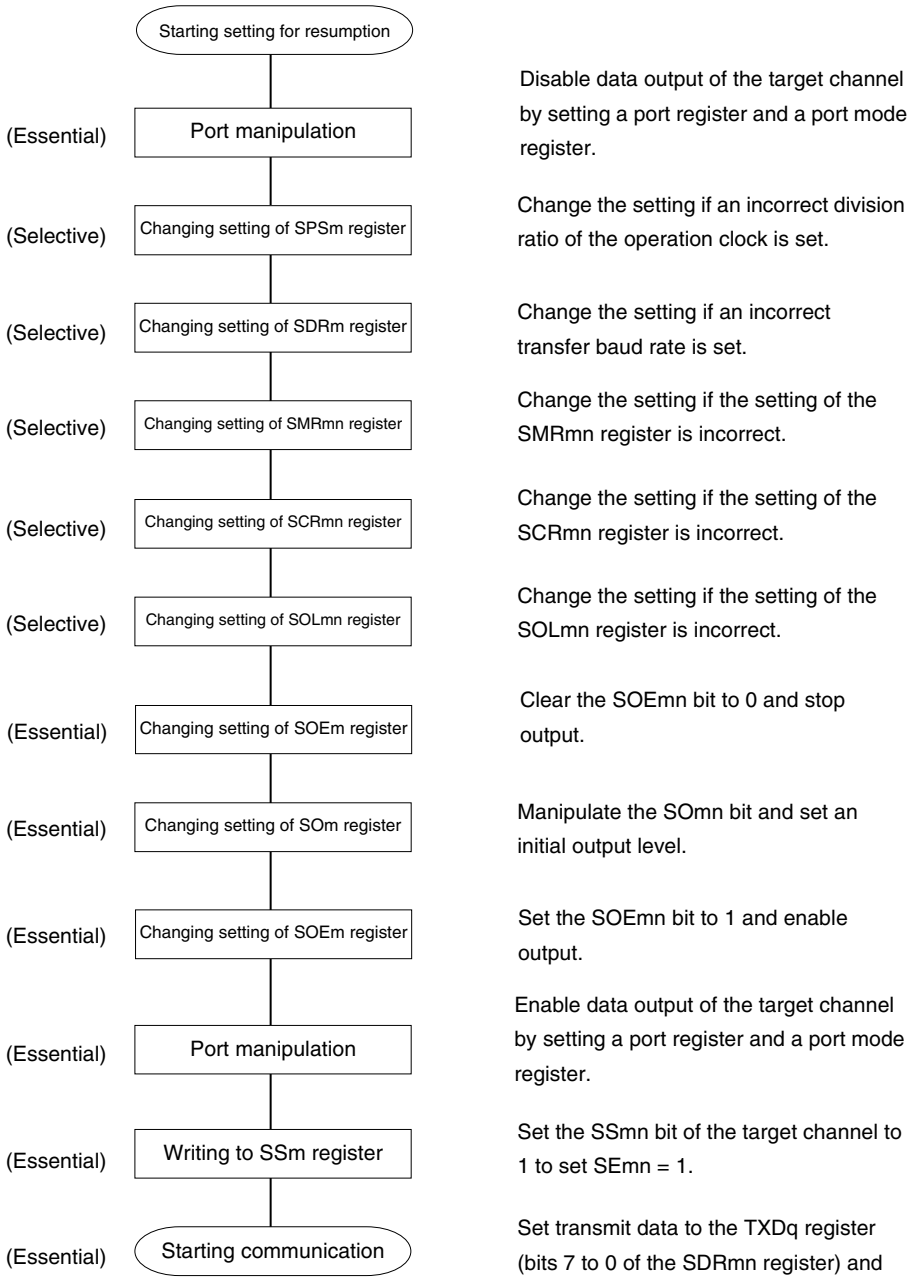
Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 11-72. Procedure for Stopping UART Transmission



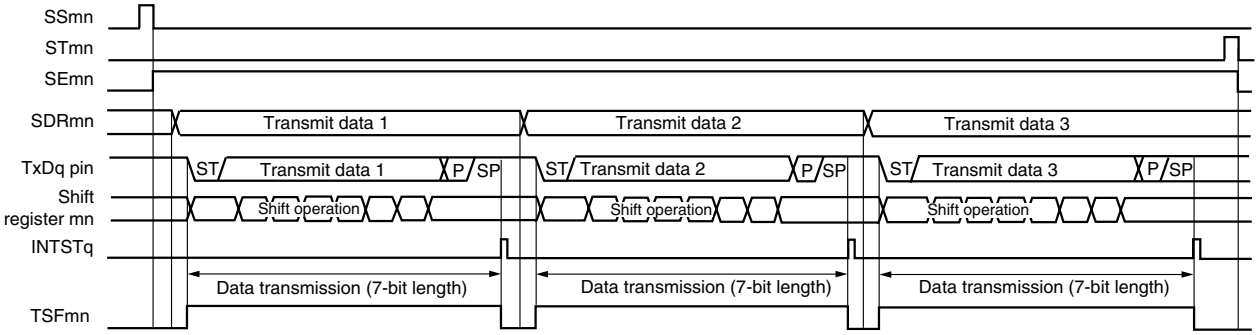
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-73 Procedure for Resuming UART Transmission**).

Figure 11-73. Procedure for Resuming UART Transmission



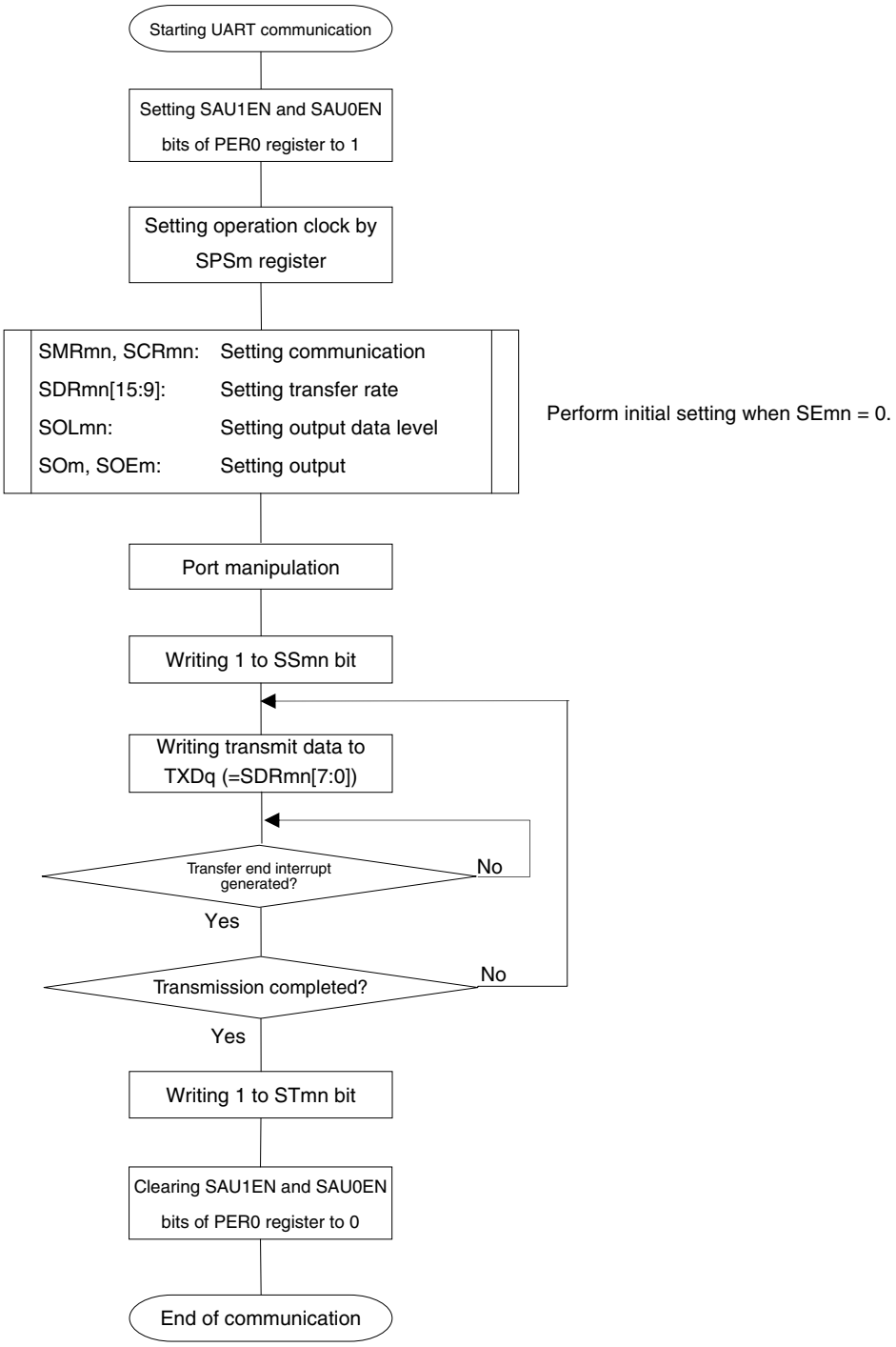
(3) Processing flow (in single-transmission mode)

Figure 11-74. Timing Chart of UART Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10,
q: UART number (q = 0 to 2)

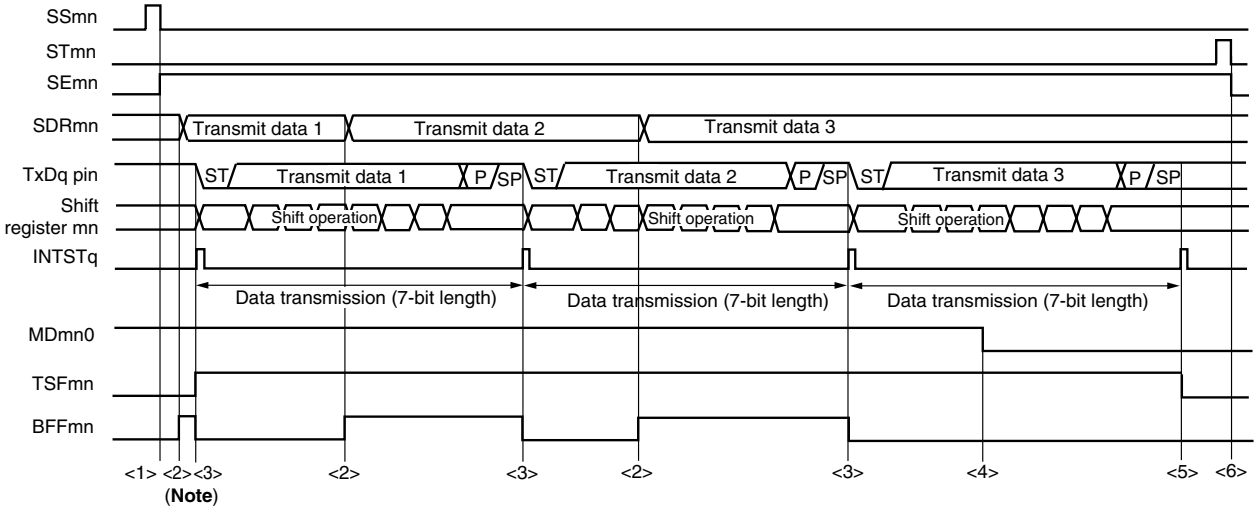
Figure 11-75. Flowchart of UART Transmission (in Single-Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 11-76. Timing Chart of UART Transmission (in Continuous Transmission Mode)

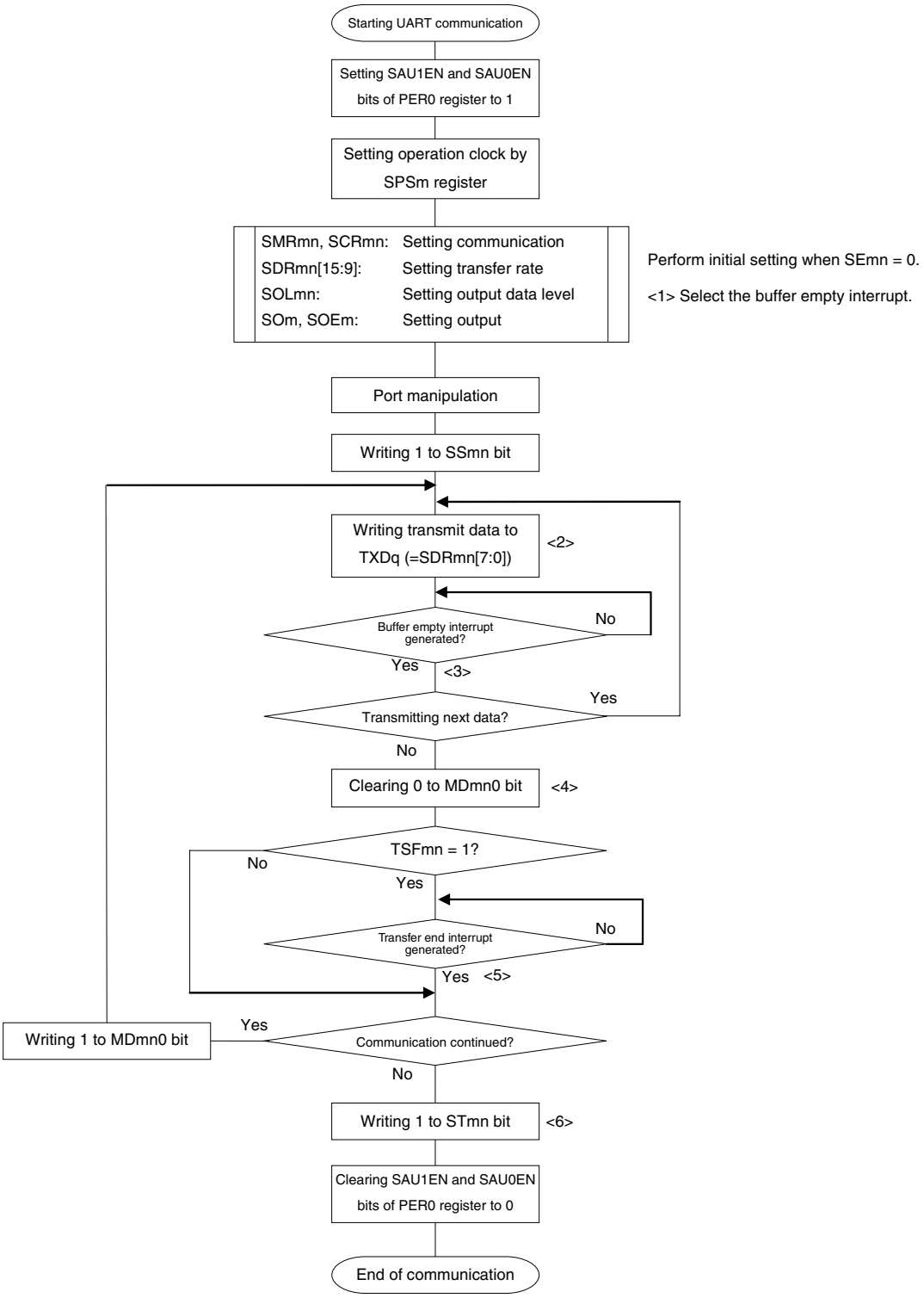


Note When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

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, 2), mn = 00, 02, 10,
q: UART number (q = 0 to 2)

Figure 11-77. Flowchart of UART Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 11-76 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

11.6.2 UART reception

UART reception is an operation wherein the 78K0R/Kx3-C 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
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1
Pins used	RxD0	RxD1	RxD2
Interrupt	INTSR0	INTSR1	INTSR2
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error interrupt	INTSRE0	INTSRE1	INTSRE2
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)		
Transfer data length	5, 7, or 8 bits		
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps]Note		
Data phase	Forward output (default: high level) Reverse output (default: low level)		
Parity bit	The following selectable • No parity bit (no parity check) • Appending 0 parity (no parity check) • Appending even parity • Appending odd parity		
Stop bit	Appending 1 bit		
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**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

(1) Register setting

Figure 11-78. Example of Contents of Registers for UART Reception of UART (UART0, UART1, UART2) (1/2)

(a) Serial output register m (SOm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm						CKOm2	CKOm1	CKOm0						SOm2	SOm1	SOm0
	0	0	0	0	1	×	×	×	0	0	0	0	1	×	×	×

(b) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm														SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	×	×	×

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3 Note	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	0/1	×

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISm0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

0: Forward (normal) reception
1: Reverse reception

Interrupt sources of channel r
0: Transfer end interrupt

(e) Serial mode register mr (SMRmr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmr	CKSmr	CCSmr						STSmr		SISmr0				MDmr2	MDmr1	MDmr0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0/1

Same setting value as CKSmn

Interrupt sources of channel r
0: Transfer end interrupt
1: Buffer empty interrupt

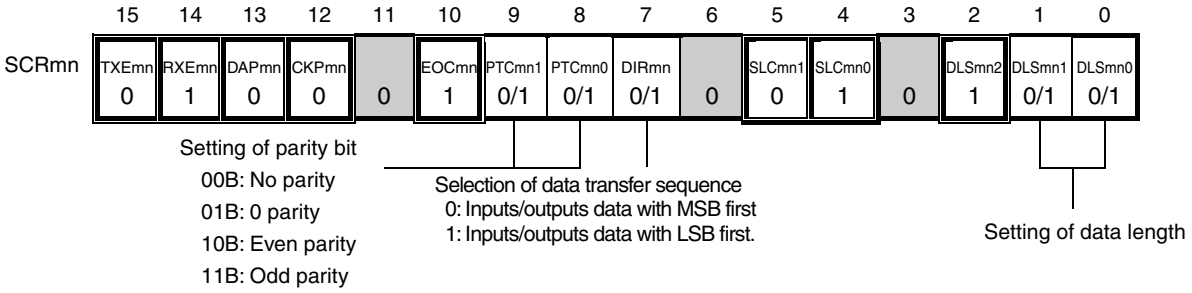
Note Serial array unit 0 only.

Caution For the UART reception, be sure to set SMRmr of channel r that is to be paired with channel n.

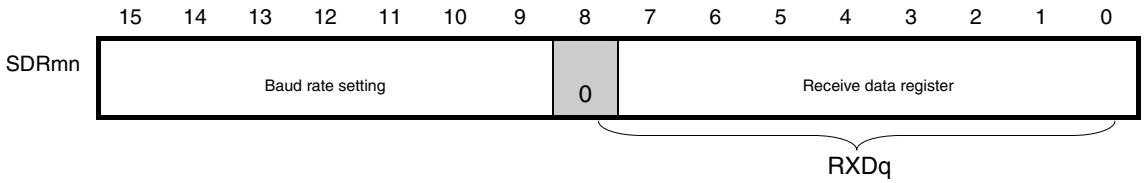
Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11,
r: Channel number (r = n – 1), q: UART number (q = 0 to 3)
□: Setting is fixed in the UART reception mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-78. Example of Contents of Registers for UART Reception of UART (UART0, UART1, UART2) (2/2)

(f) Serial communication operation setting register mn (SCRmn)



(g) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



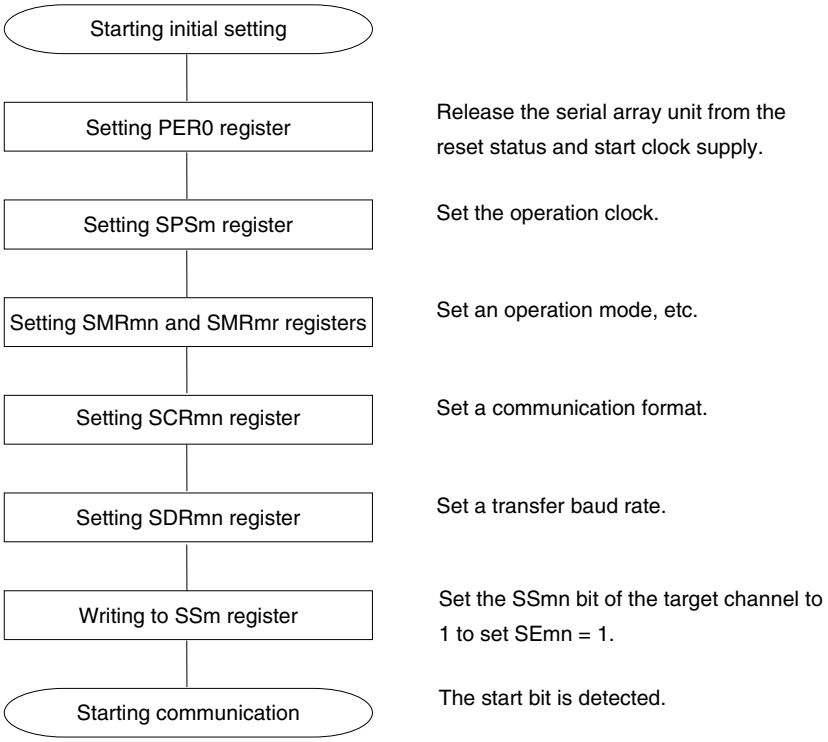
Note Serial array unit 0 only.

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), mn = 01, 03, 11,
r: Channel number (r = n – 1), q: UART number (q = 0 to 3)
□: Setting is fixed in the UART reception mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 11-79. Initial Setting Procedure for UART Reception



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

Figure 11-80. Procedure for Stopping UART Reception

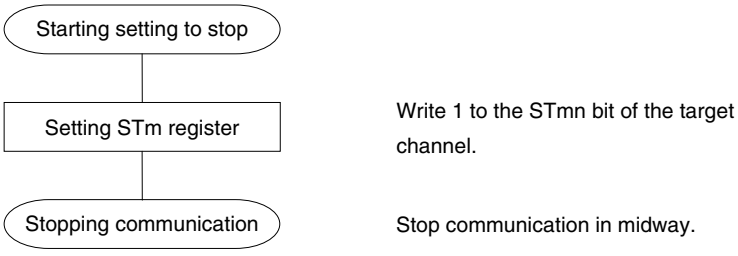
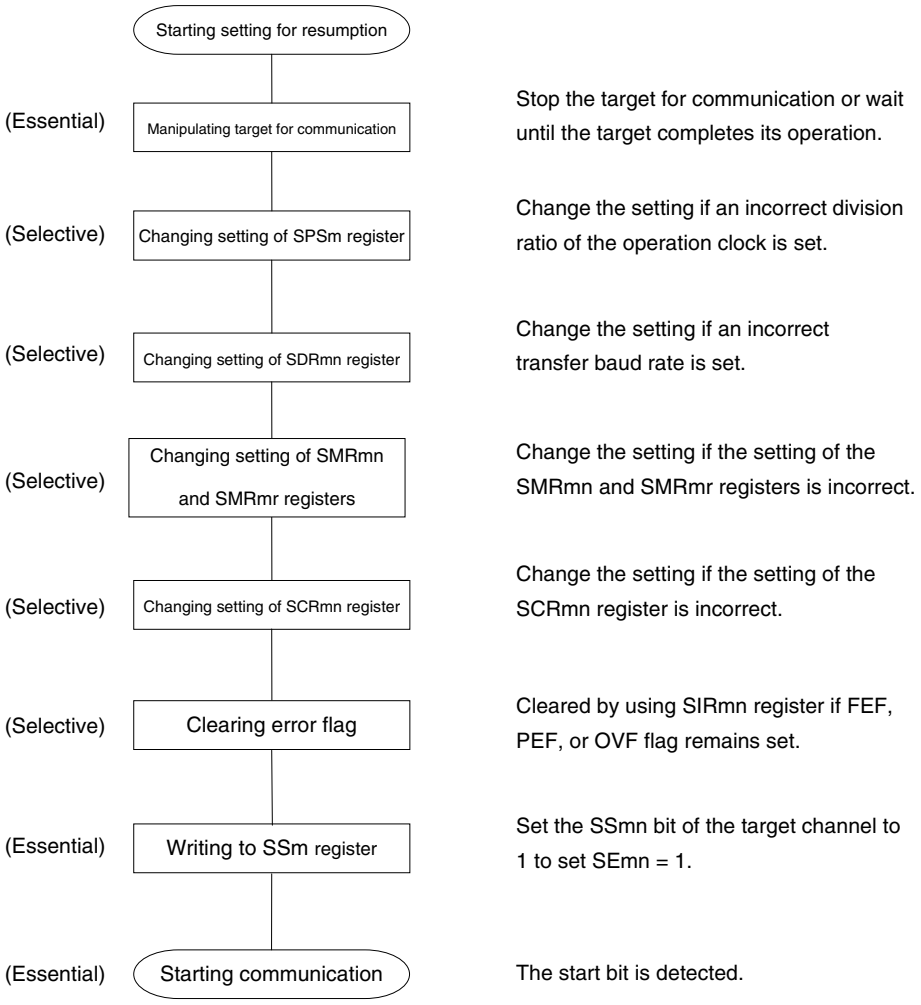
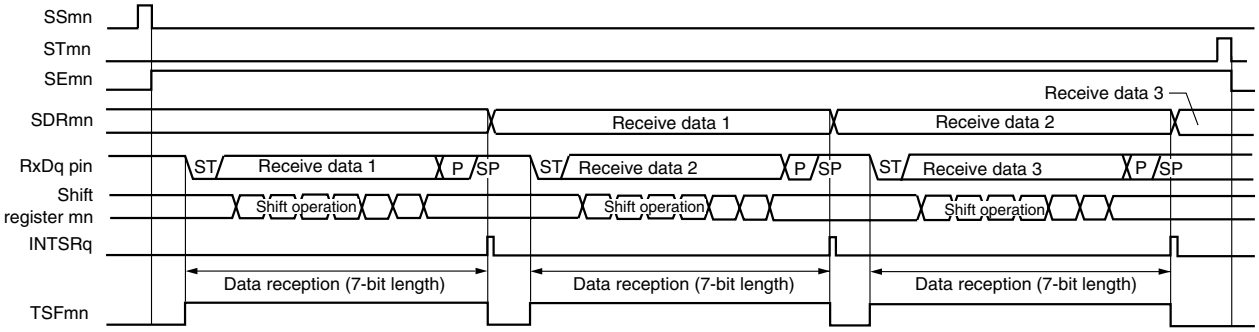


Figure 11-81. Procedure for Resuming UART Reception



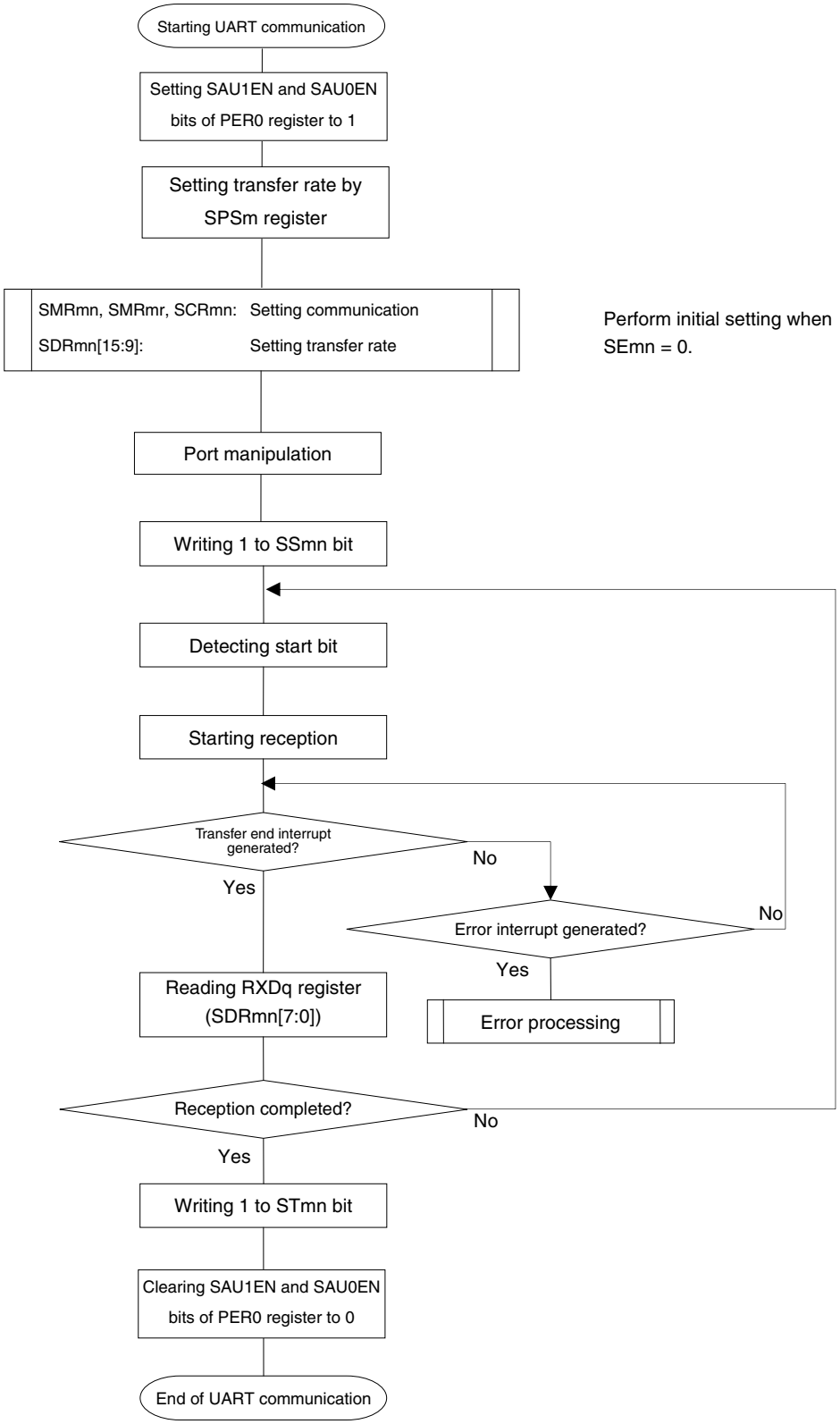
(3) Processing flow

Figure 11-82. Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11,
q: UART number (q = 0 to 2)

Figure 11-83. Flowchart of UART Reception



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

11.6.3 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0, UART1, UART2) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting 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), mn = 00 to 03, 10, 11

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11-3. Selection of Operation Clock

SMRmn Register	SPSm Register								Operation Clock (f_{MCK})[Note 1]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	313 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0[Note 2], Setting prohibited if m = 1	
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	313 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0[Note 2], Setting prohibited if m = 1	
Other than above									Setting prohibited	

Notes 1. When changing the clock selected for f_{CLK} (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 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAU0 can be used at the fixed division ratio of the subsystem clock regardless of the f_{CLK} frequency by setting TAU0 and SAU0 as follows.

<TAU0> Select $f_{SUB}/4$ as the input clock of channel 2 of TAU0. (Set TIS02 to 1.)

<SAU0> Select INTTM02 by using the SPS0 register.

However, when changing f_{CLK}, SAU0 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), mn = 00 to 03, 10, 11

(2) Baud rate error during transmission

The baud rate error of UART (UART0, UART1, UART2) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 − 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 20 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 20 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	64	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^8$	64	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^7$	64	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^6$	64	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^5$	64	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^4$	64	9615.38 bps	+0.16 %
19200 bps	$f_{CLK}/2^3$	64	19230.8 bps	+0.16 %
31250 bps	$f_{CLK}/2^3$	39	31250.0 bps	±0.0 %
38400 bps	$f_{CLK}/2^2$	64	38461.5 bps	+0.16 %
76800 bps	$f_{CLK}/2$	64	76923.1 bps	+0.16 %
153600 bps	f_{CLK}	64	153846 bps	+0.16 %
312500 bps	f_{CLK}	31	312500 bps	±0.0 %

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0, UART1, UART2) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

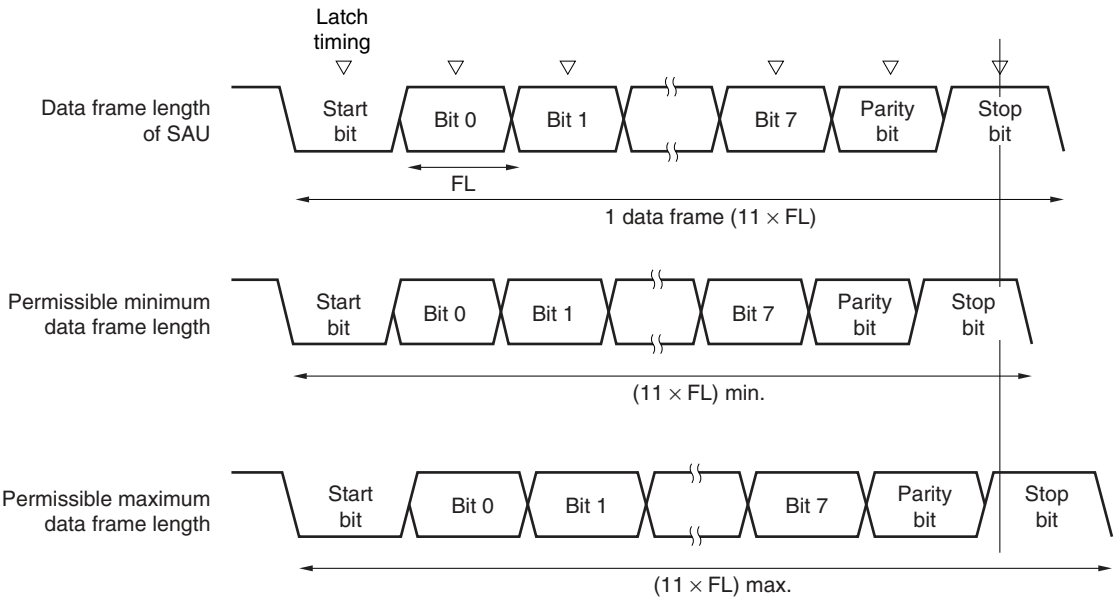
Brate: Calculated baud rate value at the reception side (See **11.6.3 (1) Baud rate calculation expression**.)

k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

Figure 11-84. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 11-84, 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.

11.6.4 Procedure for processing errors that occurred during UART (UART0, UART1, UART2) communication

The procedure for processing errors that occurred during UART (UART0, UART1, UART2) communication is described in Figures 11-85 and 11-86.

Figure 11-85. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 11-86. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operating.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

11.7 Operation of Simplified I²C (IIC10, IIC20) 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[Note] and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Overrun error
- Parity error (ACK error)

* [Functions not supported by simplified I²C]

- 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 **11.7.3 (2) Processing flow** for details.

Remarks 1. To use the full-function I²C bus, see **CHAPTER 12 SERIAL INTERFACE IICA**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10

The channels supporting simplified I²C (IIC10, IIC20) are channel 2 of SAU0 and channel 0 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	–
	1	CSI01		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–

Simplified I²C (IIC10, IIC20) performs the following four types of communication operations.

- Address field transmission (See **11.7.1**.)
- Data transmission (See **11.7.2**.)
- Data reception (See **11.7.3**.)
- Stop condition generation (See **11.7.4**.)

11.7.1 Address field transmission

Address field transmission is a transmission operation that first executes in I^2C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Simplified I^2C	IIC10	IIC20
Target channel	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	SCL10, SDA10[Note]	SCL20, SDA20[Note]
Interrupt	INTIIC10	INTIIC20
	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 frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)	
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^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POM03, POM143 = 1) for the port output mode registers (POM0, 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, POM142 = 1) also for the clock input/output pins (SCL10, SCL20) (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, 2), mn = 02, 10

(1) Register setting

Figure 11-87. Example of Contents of Registers for Address Field Transmission of Simplified I^2C (IIC10, IIC20)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 Note 1 0/1	CKOm1 ×	CKOm0 Note 2 0/1	0	0	0	0	1	SOm2 Note 1 0/1	SOm1 ×	SOm0 Note 2 0/1

Start condition is generated by manipulating the SOmn bit.

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 Note 1 0/1	SOEm1 ×	SOEm0 Note 2 0/1

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 to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 Note 1 0/1	SSm1 ×	SSm0 Note 2 0/1

(d) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISm0 0	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt sources of channel n
0: Transfer end interrupt

(e) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0	SLCmn0 1	0	DLSmn2 1	DLSmn1 1	DLSmn0 1

Setting of parity bit
00B: No parity

Setting of stop bit
01B: Appending 1 bit (ACK)

(f) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting (address + R/W)							

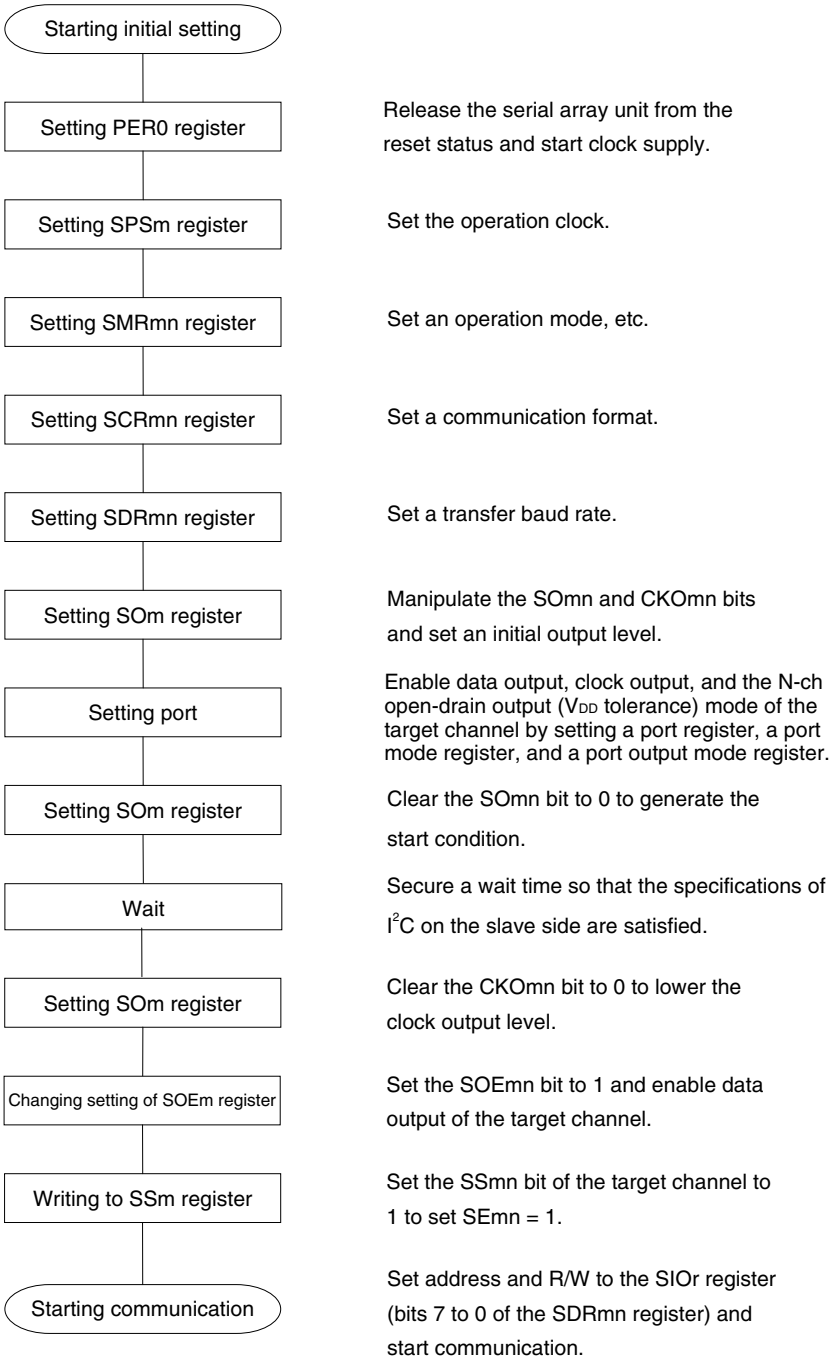
SIOr

Notes 1. Serial array unit 0 only.
2. Serial array unit 1 only.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)
□ : Setting is fixed in the IIC mode, ■ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

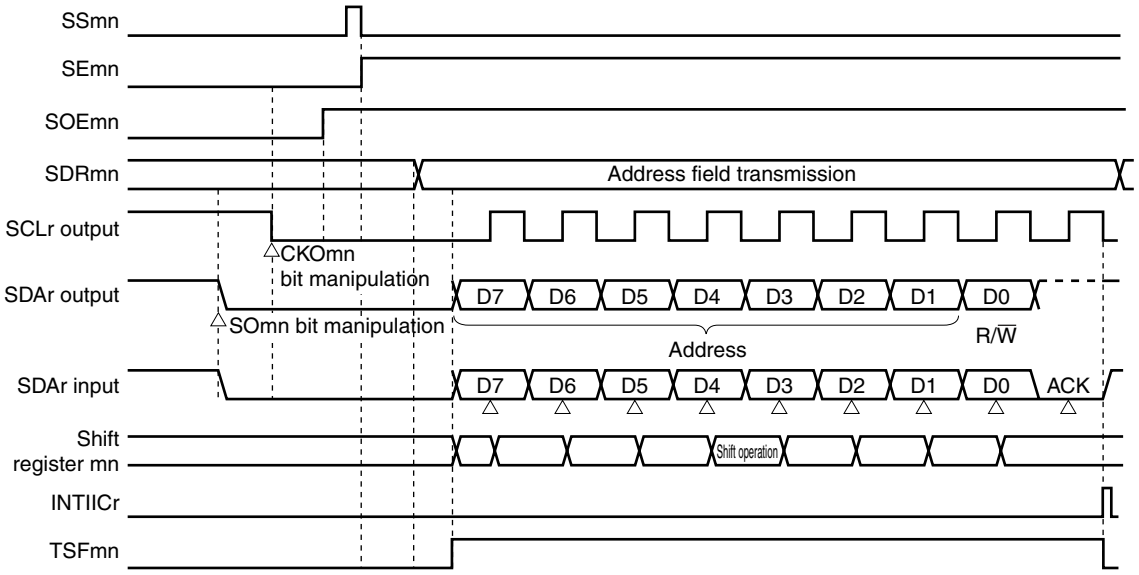
Figure 11-88. Initial Setting Procedure for Address Field Transmission



Caution After setting the SAUmEN bit of peripheral enable register 0 (PER0) to 1, be sure to set serial clock select register m (SPSm) after 4 or more f_{CLK} clocks have elapsed.

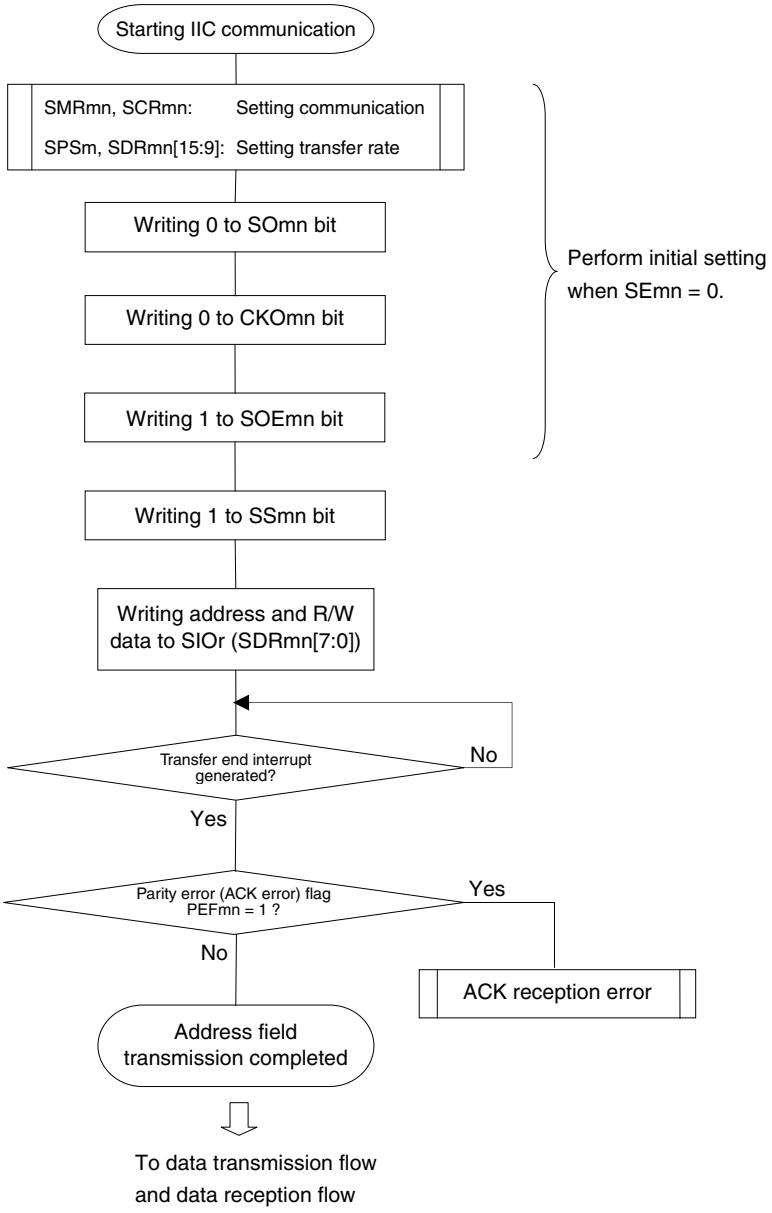
(3) Processing flow

Figure 11-89. Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)

Figure 11-90. Flowchart of Address Field Transmission



11.7.2 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^2C	IIC10	IIC20
Target channel	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	SCL10, SDA10[Note]	SCL20, SDA20[Note]
Interrupt	INTIIC10	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Parity error detection flag (PEFmn)	
Transfer data length	8 bits	
Transfer rate	Max. f_{MCK}/4 [Hz] (SDRmn [15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)	
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^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POM03, POM143 = 1) for the port output mode registers (POM0, 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, POM142 = 1) also for the clock input/output pins (SCL10, SCL20) (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, 2), mn = 02, 10

(1) Register setting

Figure 11-91. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC10, IIC20) (1/2)

(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	0	0	0	0	1	CKOm2 Note 1: 0/1 Note 3	CKOm1: ×	CKOm0 Note 2: 0/1 Note 3	0	0	0	0	1	SOm2 Note 1: 0/1 Note 3	SOm1: ×	SOm0 Note 2: 0/1 Note 3

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.

0: Serial data output value is "0"
1: Serial data output value is "1"

(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	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 Note 1: 1	SOEm1: ×	SOEm0 Note 2: 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	0	0	0	0	0	0	0	0	0	0	0	0	SSm3: ×	SSm2 Note 1: 0/1	SSm1: ×	SSm0 Note 2: 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: 0/1	CCSmn: 0	0	0	0	0	0	STSmn: 0	0	SISm0: 0	1	0	0	MDmn2: 1	MDmn1: 0	MDmn0: 0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Notes 1. Serial array unit 0 only.
2. Serial array unit 1 only.
3. The value varies depending on the communication data during communication operation.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)
□: 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

Figure 11-91. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC10, IIC20) (2/2)

(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: SIOr)

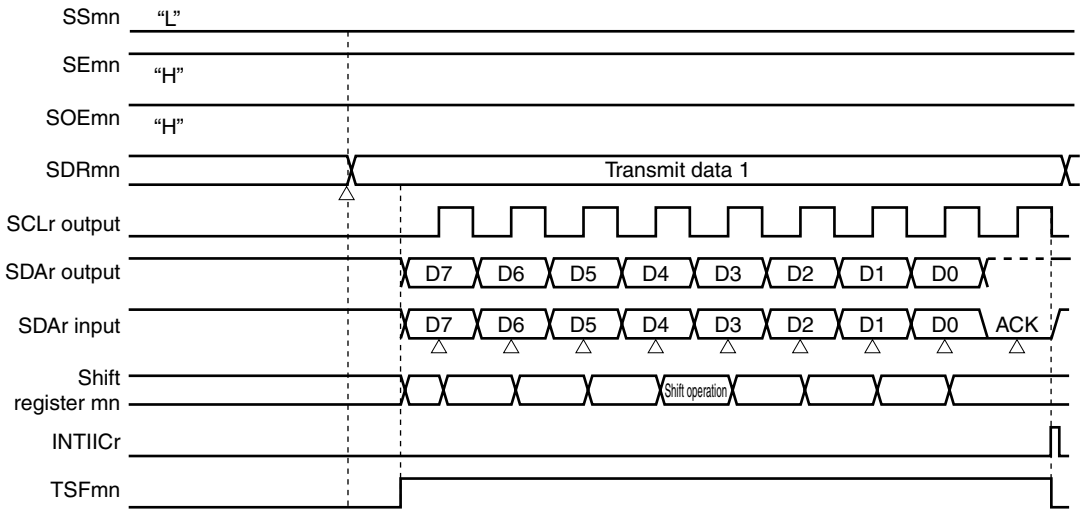
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting							

SIOr

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)
□: 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

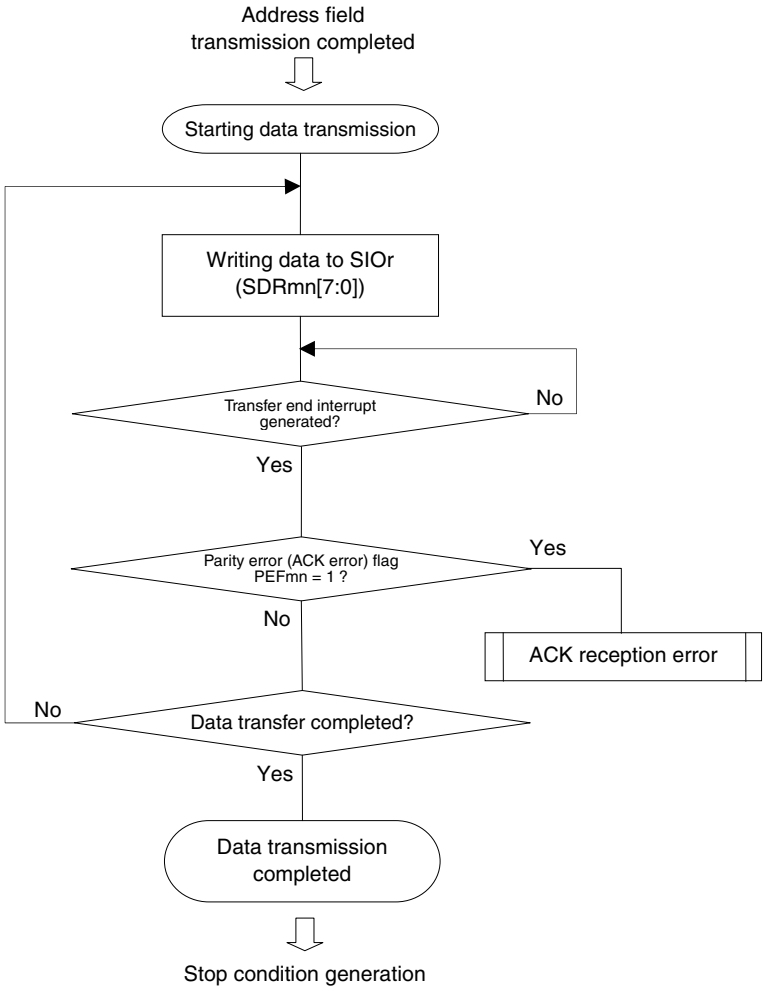
(2) Processing flow

Figure 11-92. Timing Chart of Data Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)

Figure 11-93. Flowchart of Data Transmission



11.7.3 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^2C	IIC10	IIC20
Target channel	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	SCL10, SDA10[Note]	SCL20, SDA20[Note]
Interrupt	INTIIC10	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	8 bits	
Transfer rate	Max. f_{MCK}/4 [Hz] (SDRmn [15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)	
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^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POM03, POM143 = 1) for the port output mode registers (POM0, 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, POM142 = 1) also for the clock input/output pins (SCL10, SCL20) (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, 2), mn = 02, 10

(1) Register setting

Figure 11-94. Example of Contents of Registers for Data Reception of Simplified I²C (IIC10, IIC20) (1/2)

(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	0	0	0	0	1	CKOm2 Note 1: 0/1 Note 3	CKOm1: ×	CKOm0 Note 2: 0/1 Note 3	0	0	0	0	1	SOm2 Note 1: 0/1 Note 3	SOm1: ×	SOm0 Note 2: 0/1 Note 3

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.

0: Serial data output value is "0"
1: Serial data output value is "1"

(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	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 Note 1: 0/1	SOEm1: ×	SOEm0 Note 2: 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	0	0	0	0	0	0	0	0	0	0	0	0	SSm3: ×	SSm2 Note 1: 0/1	SSm1: ×	SSm0 Note 2: 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: 0/1	CCSmn: 0	0	0	0	0	0	STSmn: 0	0	SISmn0: 0	1	0	0	MDmn2: 1	MDmn1: 0	MDmn0: 0

Notes 1. Serial array unit 0 only.
2. Serial array unit 1 only.
3. The value varies depending on the communication data during communication operation.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)
□: 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

Figure 11-94. Example of Contents of Registers for Data Reception of Simplified I^2C (IIC10, IIC20) (2/2)

(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: SIOr)

	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)							

SIOr

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)

□: Setting is fixed in the IIC mode, ■: Setting disabled (set to the initial value)

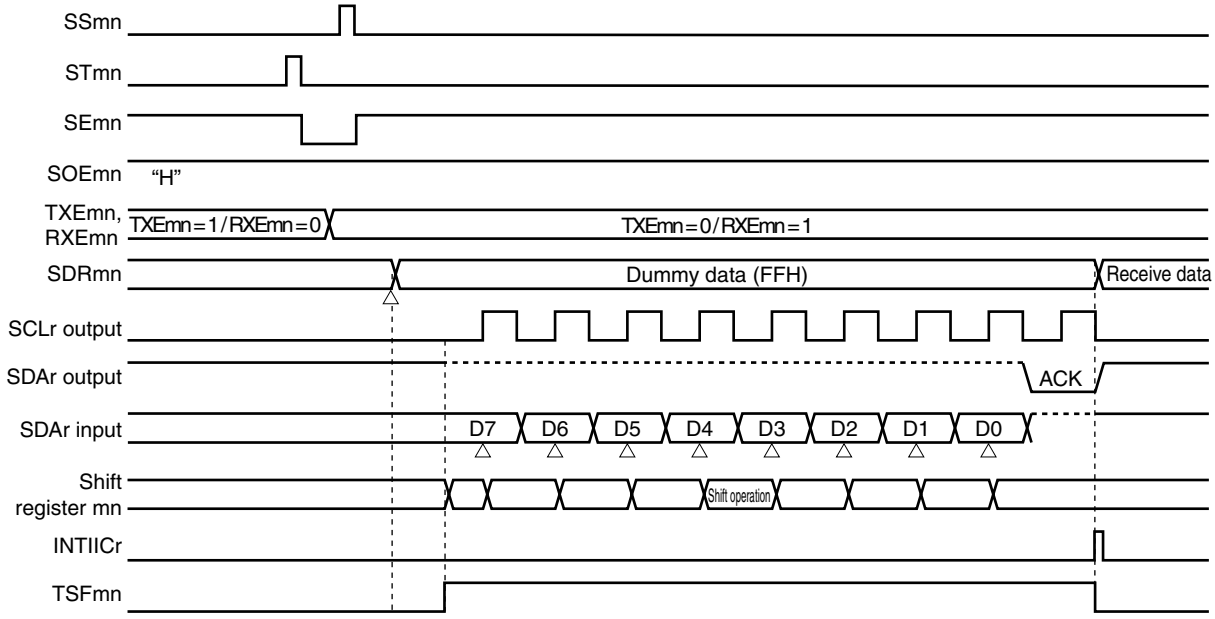
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

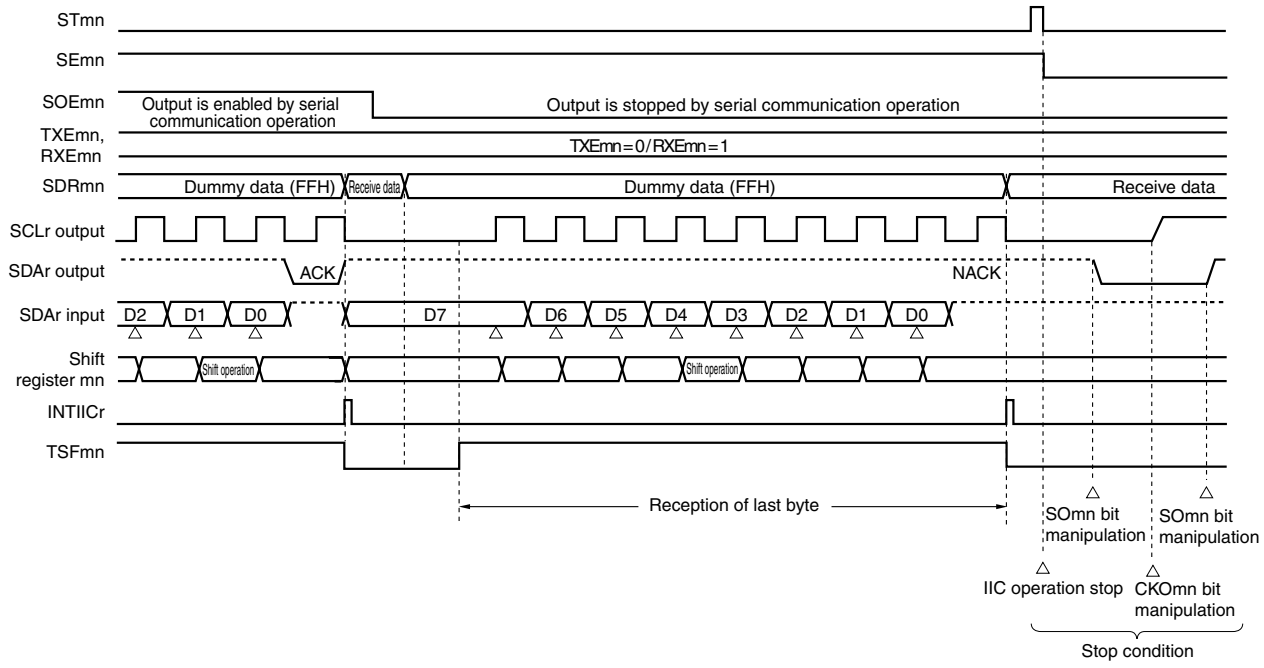
(2) Processing flow

Figure 11-95. Timing Chart of Data Reception

(a) When starting data reception

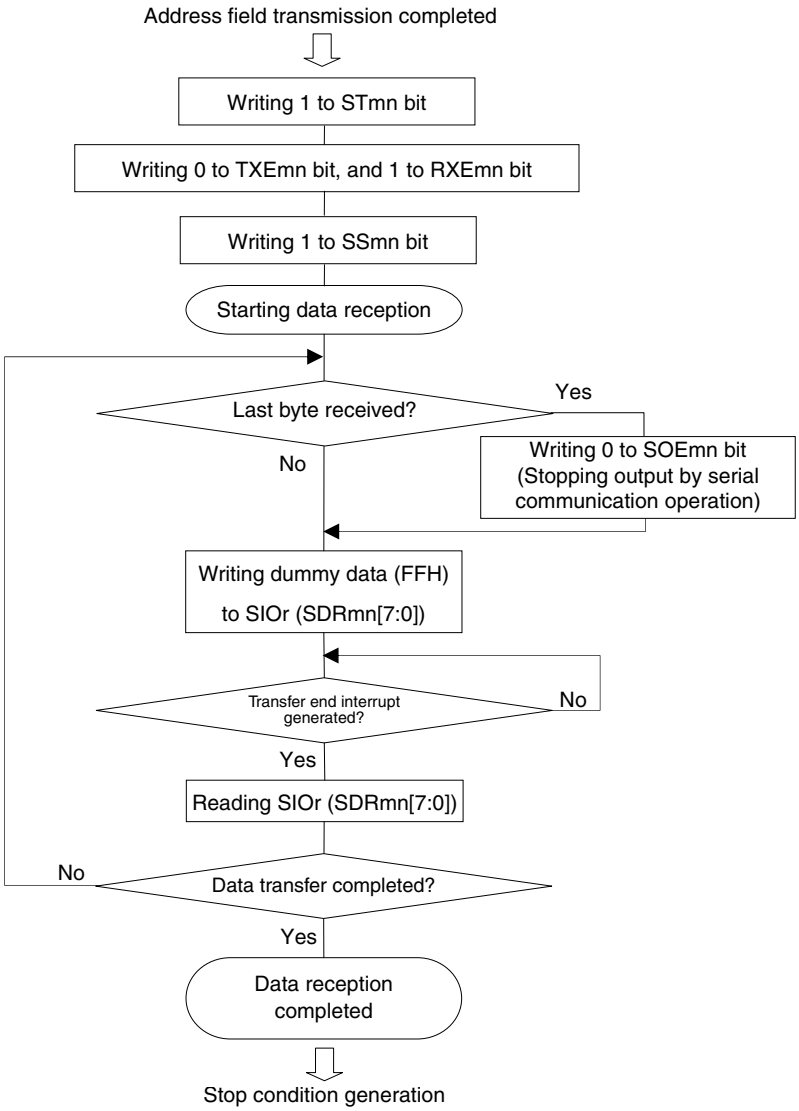


(b) When receiving last data



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10, r: IIC number (r = 10, 20)

Figure 11-96. Flowchart of Data Reception



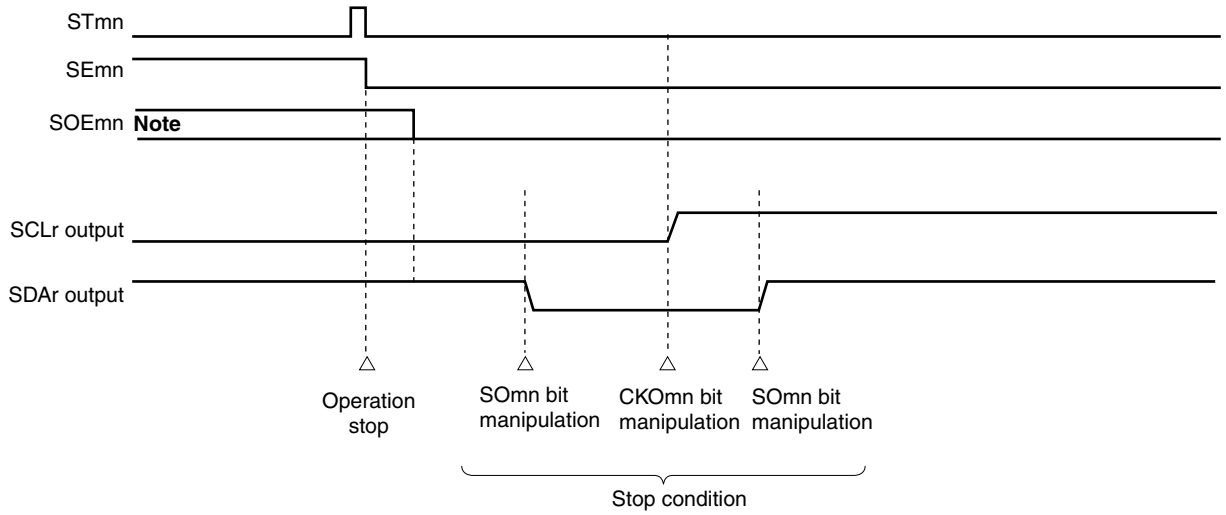
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting "1" to the STmn bit to stop operation and generating a stop condition.

11.7.4 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

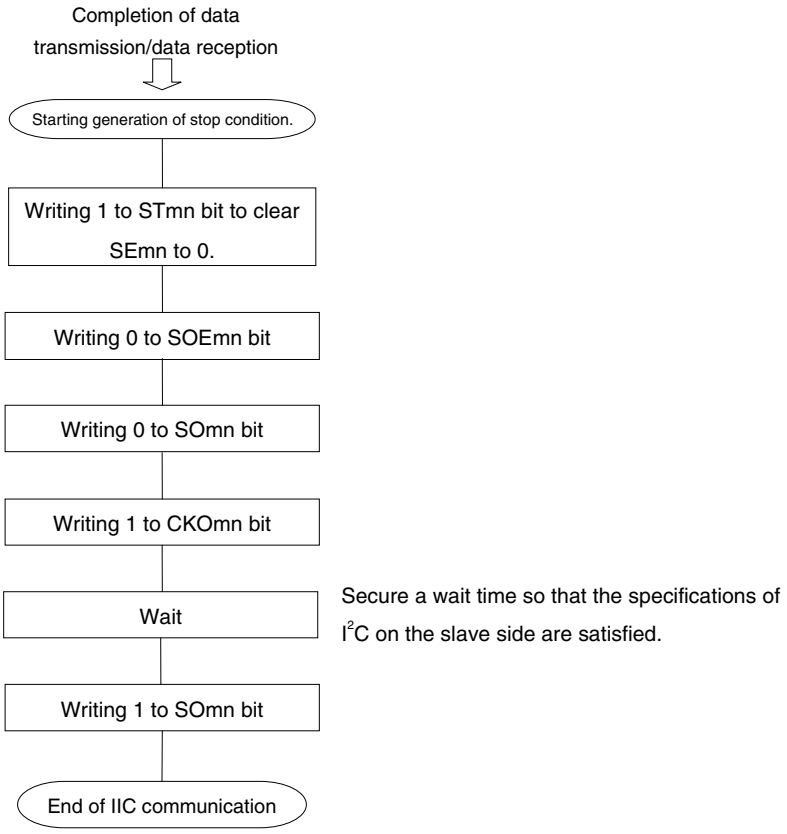
Figure 11-97. Timing Chart of Stop Condition Generation



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, 2), mn = 02, 10, r: IIC number (r = 10, 20)

Figure 11-98. Flowchart of Stop Condition Generation



11.7.5 Calculating transfer rate

The transfer rate for simplified I^2C (IIC10, IIC20) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

Caution Setting SDRmn[15:9] = 0000000B is prohibited when simplified I^2C is used. Setting SDRmn[15:9] = 0000001B or more.

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, 2), mn = 02, 10

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11-4. Selection of Operation Clock

SMRmn Register	SPSm Register								Operation Clock (f_{MCK})[Note 1]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	313 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0[Note 2], Setting prohibited if m = 1	
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	313 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0[Note 2], Setting prohibited if m = 1	
Other than above									Setting prohibited	

Notes 1. When changing the clock selected for f_{CLK} (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 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAU0 can be used at the fixed division ratio of the subsystem clock regardless of the f_{CLK} frequency by setting TAU0 and SAU0 as follows.

<TAU0> Select $f_{SUB}/4$ as the input clock of channel 2 of TAU0. (Set TIS02 to 1.)

<SAU0> Select INTTM02 by using the SPS0 register.

However, when changing f_{CLK}, SAU0 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, 2), mn = 02, 10

Here is an example of setting an IIC transfer rate where $f_{MCK} = f_{CLK}$ = 20 MHz.

IIC Transfer Mode (Desired Transfer Rate)	f_{CLK} = 20 MHz			
	Operation Clock (f_{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%

11.7.6 Procedure for processing errors that occurred during simplified I²C (IIC10, IIC20) communication

The procedure for processing errors that occurred during simplified I²C (IIC10, IIC20) communication is described in Figures11-99 and 11-100.

Figure 11-99. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 11-100. Processing Procedure in Case of Parity Error (ACK error) in Simplified I²C Mode

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	Slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates stop condition.		
Creates start condition.		
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 10, 20)

11.8 Relationship Between Register Settings and Pins

Tables 11-5 to 11-10 show the relationship between register settings and pins for each channel of serial array units 0 and 1.

Table 11-5. Relationship Between Register Settings and Pins

(Channel 0 of Unit 0: CSI00, UART0 Transmission)

SE00 Note 1	MD002	MD001	SOE00	SO00	CKO00	TXE00	RXE00	PM10	P10	PM11 Note 2	P11 Note 2	PM12	P12	Operation Mode	Pin Function		
															$\overline{\text{SCK00}}$/P10	SI00/RxD0/P11 Note 2	SO00/TxD0/P12
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	P10	P11	P12
	0	1														P11/ RxD0	
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	Slave CSI00 reception	$\overline{\text{SCK00}}$ (input)	SI00	P12
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	Slave CSI00 transmission	$\overline{\text{SCK00}}$ (input)	P11	SO00
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	Slave CSI00 transmission/ reception	$\overline{\text{SCK00}}$ (input)	SI00	SO00
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	Master CSI00 reception	$\overline{\text{SCK00}}$ (output)	SI00	P12
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	Master CSI00 transmission	$\overline{\text{SCK00}}$ (output)	P11	SO00
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	Master CSI00 transmission/ reception	$\overline{\text{SCK00}}$ (output)	SI00	SO00
	0	1	1	0/1 Note 4	1	1	0	× Note 3	× Note 3	× Note 3	× Note 3	0	1	UART0 transmission Note 5	P10	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 11-6**). 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 **11.3 (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 11-6**).

Remark X: Don't care

Table 11-6. Relationship Between Register Settings and Pins (Channel 1 of Unit 0: CSI01, UART0 Reception)

SE 01 Note 1	MD 012	MD 011	SOE 01	SO01	CKO 01	TXE 01	RXE 01	PM 43	P43	PM44	P44	PM 45	P45	PM 11 Note 2	P11 Note 2	Operation Mode	Pin Function			
																	$\overline{SCK01}$/ P43	SI01/P44	SO01/ P45	SI00/ RxD0/ P11 Note 2
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	P43	P44	P45	SI00/P11
	0	1																		
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	× Note 3	× Note 3	Slave CSI01 reception	$\overline{SCK01}$ (input)	SI01	P45	SI00/P11
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	× Note 3	× Note 3	Slave CSI01 transmission	$\overline{SCK01}$ (input)	P44	SO01	SI00/P11
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	× Note 3	× Note 3	Slave CSI01 transmission /reception	$\overline{SCK01}$ (input)	SI01	SO01	SI00/P11
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	× Note 3	× Note 3	Master CSI01 reception	$\overline{SCK01}$ (output)	SI01	P45	SI00/P11
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	× Note 3	× Note 3	Master CSI01 transmission	$\overline{SCK01}$ (output)	P44	SO01	SI00/P11
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	× Note 3	× Note 3	Master CSI01 transmission /reception	$\overline{SCK01}$ (output)	SI01	SO01	SI00/P11
	0	1	0	1	1	0	1	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	1	×	UART0 reception Notes 5, 6	P43	P44	P45	RxD0

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. In this case, set channel 0 of unit 0 to operation stop mode or UART0 transmission (refer to **Table 11-5**).
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.

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 **11.3 (12) Serial output register m (SOm)**.

5. When using UART0 transmission and reception in a pair, set channel 0 of unit 0 to UART0 transmission (refer to **Table 11-5**).

6. The SMR00 register of channel 0 of unit 0 must also be set during UART0 reception. For details, refer to **11.6.2 (1) Register setting**.

Remark X: Don't care

Table 11-7. Relationship Between Register Settings and Pins
(Channel 2 of Unit 0: CSI10, UART1 Transmission, IIC10)

SE02 Note 1	MD022	MD021	SOE02	SO02	CKO02	TXE02	RXE02	PM04	P04	PM03 Note 2	P03 Note 2	PM02	P02	Operation Mode	Pin Function		
															$\overline{SCK10}$/SCL10/P04	SI10/SDA10/RxD1/P03 Note 2	SO10/TxD1/P02
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	P04	P03	P02
	0	1														P03/RxD1	
	1	0														P03	
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	Slave CSI10 reception	$\overline{SCK10}$ (input)	SI10	P02
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	Slave CSI10 transmission	$\overline{SCK10}$ (input)	P03	SO10
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	Slave CSI10 transmission /reception	$\overline{SCK10}$ (input)	SI10	SO10
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	Master CSI10 reception	$\overline{SCK10}$ (output)	SI10	P02
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	Master CSI10 transmission	$\overline{SCK10}$ (output)	P03	SO10
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	Master CSI10 transmission /reception	$\overline{SCK10}$ (output)	SI10	SO10
	0	1	1	0/1 Note 4	1	1	0	× Note 3	× Note 3	× Note 3	× Note 3	0	1	UART1 transmission Note 5	P04	P03/RxD1	TxD1
0	1	0	0	0/1 Note 6	0/1 Note 6	0	0	0	1	0	1	× Note 3	× Note 3	IIC10 start condition	SCL10	SDA10	P02
						1	0										
						0	1										
1			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC10 address field transmission	SCL10	SDA10	P02
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC10 data transmission	SCL10	SDA10	P02
			1	0/1 Note 4	0/1 Note 4	0	1	0	1	0	1	× Note 3	× Note 3	IIC10 data reception	SCL10	SDA10	P02
0			0	0/1 Note 7	0/1 Note 7	0	0	0	1	0	1	× Note 3	× Note 3	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.

2. When channel 3 of unit 0 is set to UART1 reception, this pin becomes an RxD1 function pin (refer to **Table 11-8**). In this case, operation stop mode or UART1 transmission must be selected for channel 2 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 **11.3 (12) Serial output register m (SOm)**.

5. When using UART1 transmission and reception in a pair, set channel 3 of unit 0 to UART1 reception (refer to **Table 11-8**).

6. 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.

7. 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 11-8. Relationship Between Register Settings and Pins (Channel 3 of Unit 0: UART1 Reception)

SE03 Note 1	MD032	MD031	TXE03	RXE03	PM03 Note 2	P03 Note 2	Operation Mode	Pin Function
								SI10/SDA10/RxD1/P03 Note 2
0	0	1	0	0	× Note 3	× Note 3	Operation stop mode	SI10/SDA10/P03 Note 2
1	0	1	0	1	1	×	UART1 reception Notes 4, 5	RxD1

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 11-7**).
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.

3. This pin can be set as a port function pin.

4. When using UART1 transmission and reception in a pair, set channel 2 of unit 0 to UART1 transmission (refer to **Table 11-7**).

5. The SMR02 register of channel 2 of unit 0 must also be set during UART1 reception. For details, refer to **11.6.2 (1) Register setting**.

Remark X: Don't care

Table 11-9. Relationship Between Register Settings and Pins
(Channel 0 of Unit 1: CSI20, UART2 Transmission, IIC20)

SE10 Note 1	MD102	MD101	SOE10	SO10	CKO10	TXE10	RXE10	PM142	P142	PM143 Note 2	P143 Note 2	PM144	P144	Operation Mode	Pin Function		
															$\overline{\text{SCK20}}$/SCL20/P142	SI20/SDA20/RxD2/P143 Note 2	SO20/TxD2/P144
0	0	0	0	1	1	0	0	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	× Note 3	Operation stop mode	P142	P143	P144
	0	1														P143/RxD2	
	1	0														P143	
1	0	0	0	1	1	0	1	1	×	1	×	× Note 3	× Note 3	Slave CSI20 reception	$\overline{\text{SCK20}}$ (input)	SI20	P144
			1	0/1 Note 4	1	1	0	1	×	× Note 3	× Note 3	0	1	Slave CSI20 transmission	$\overline{\text{SCK20}}$ (input)	P143	SO20
			1	0/1 Note 4	1	1	1	1	×	1	×	0	1	Slave CSI20 transmission/reception	$\overline{\text{SCK20}}$ (input)	SI20	SO20
			0	1	0/1 Note 4	0	1	0	1	1	×	× Note 3	× Note 3	Master CSI20 reception	$\overline{\text{SCK20}}$ (output)	SI20	P144
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	× Note 3	× Note 3	0	1	Master CSI20 transmission	$\overline{\text{SCK20}}$ (output)	P143	SO20
			1	0/1 Note 4	0/1 Note 4	1	1	0	1	1	×	0	1	Master CSI20 transmission/reception	$\overline{\text{SCK20}}$ (output)	SI20	SO20
	0	1	1	0/1 Note 4	1	1	0	× Note 3	× Note 3	× Note 3	× Note 3	0	1	UART2 transmission Note 5	P142	P143/RxD2	TxD2
0	1	0	0	0/1 Note 6	0/1 Note 6	0	0	0	1	0	1	× Note 3	× Note 3	IIC20 start condition	SCL20	SDA20	P144
						1	0										
						0	1										
1			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC20 address field transmission	SCL20	SDA20	P144
			1	0/1 Note 4	0/1 Note 4	1	0	0	1	0	1	× Note 3	× Note 3	IIC20 data transmission	SCL20	SDA20	P144
			1	0/1 Note 4	0/1 Note 4	0	1	0	1	0	1	× Note 3	× Note 3	IIC20 data reception	SCL20	SDA20	P144
0			0	0/1 Note 7	0/1 Note 7	0	0	0	1	0	1	× Note 3	× Note 3	IIC20 stop condition	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.

2. When channel 1 of unit 1 is set to UART2 reception, this pin becomes an RxD2 function pin (refer to **Table 11-10**). In this case, operation stop mode or UART2 transmission must be selected for channel 0 of unit 1.

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 **11.3 (12) Serial output register m (SOm)**.

5. When using UART2 transmission and reception in a pair, set channel 1 of unit 1 to UART2 reception (refer to **Table 11-10**).

6. 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.

7. 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 11-10. Relationship Between Register Settings and Pins (Channel 1 of Unit 1: UART2 Reception)

SE11 Note 1	MD112	MD111	TXE11	RXE11	PM143 Note 2	P143 Note 2	Operation Mode	Pin Function
								SI20/SDA20/RxD2/P143 Note 2
0	0	1	0	0	× Note 3	× Note 3	Operation stop mode	SI20/SDA20/P143
1	0	1	0	1	1	×	UART2 reception Notes 4, 5	RxD2

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 11-9**).
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 1 of unit 1 to operation stop mode.

3. This pin can be set as a port function pin.

4. When using UART2 transmission and reception in a pair, set channel 0 of unit 1 to UART2 transmission (refer to **Table 11-9**).

5. The SMR10 register of channel 0 of unit 1 must also be set during UART2 reception. For details, refer to **11.6.2 (1) Register setting**.

Remark X: Don't care

CHAPTER 12 SERIAL INTERFACE IICA

12.1 Functions of Serial Interface IICA

Serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL0) line and a serial data bus (SDA0) line.

This mode complies with the I^2C bus format and the master device can generated “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the I^2C bus.

Since the SCL0 and SDA0 pins are used for open drain outputs, serial interface IICA requires pull-up resistors for the serial clock line and the serial data bus line.

(3) Wakeup mode

The STOP mode can be released by generating an interrupt request signal (INTIICA) when an extension code from the master device or a local address has been received while in STOP mode. This can be set by using the WUP bit of IICA control register 1 (IICCTL1).

Figure 12-1 shows a block diagram of serial interface IICA.

Figure 12-1. Block Diagram of Serial Interface IICA

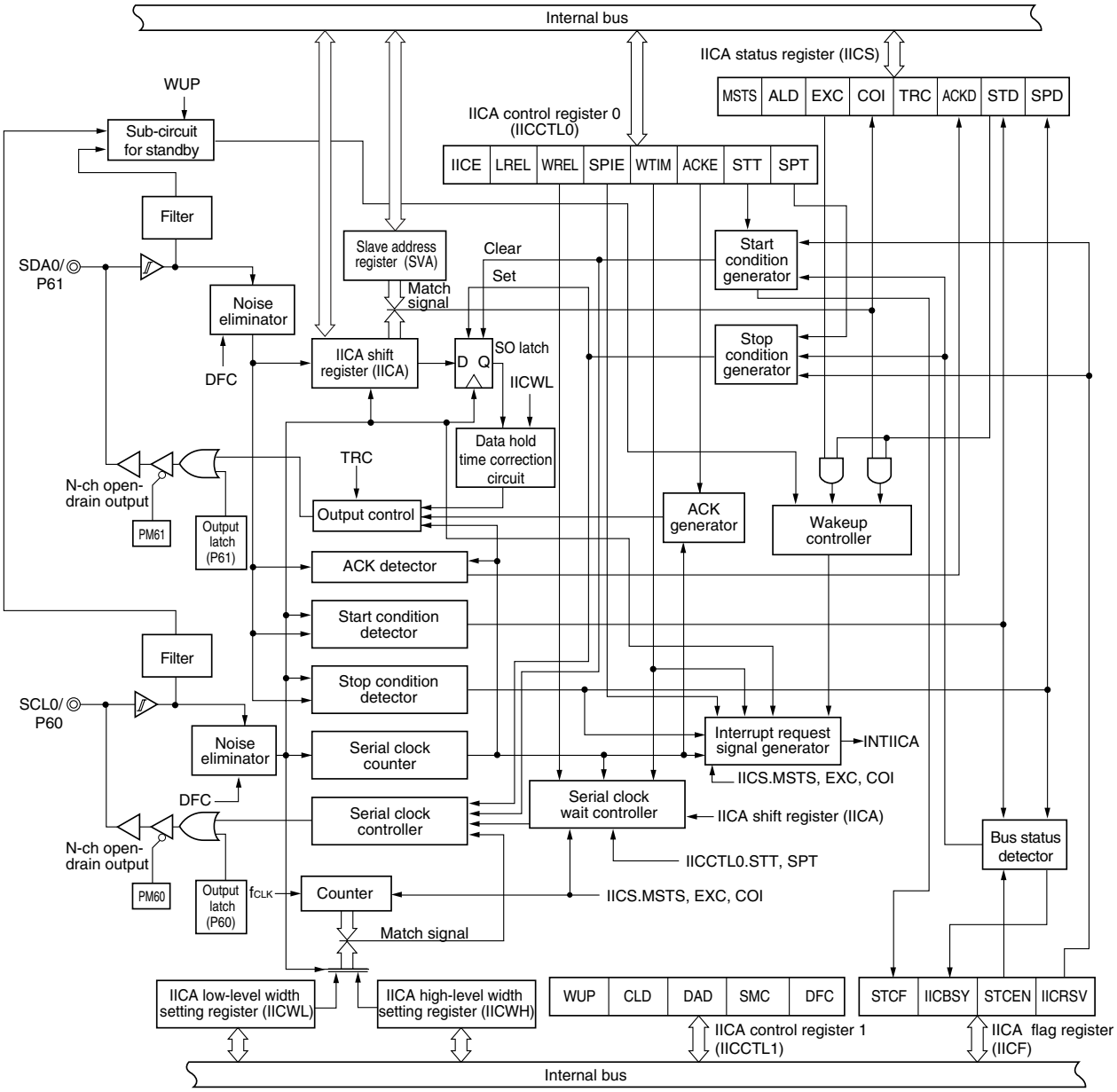
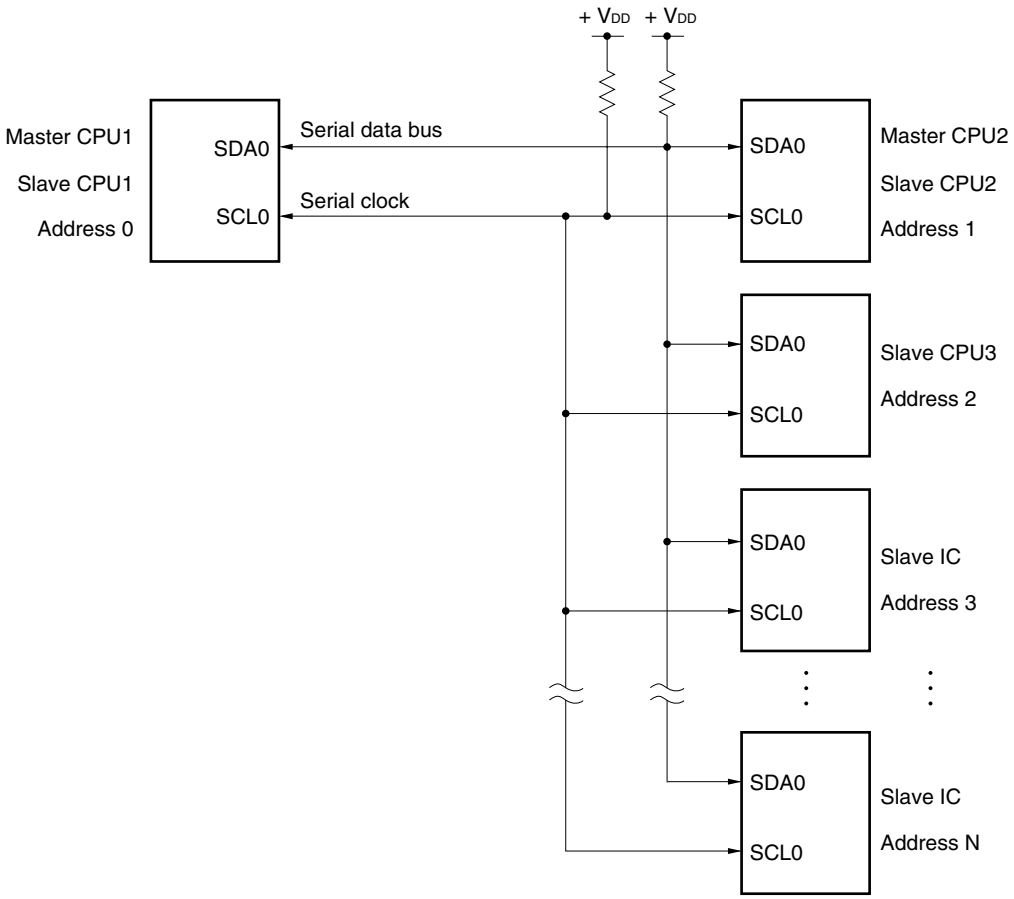


Figure 12-2 shows a serial bus configuration example.

Figure 12-2. Serial Bus Configuration Example Using I²C Bus



12.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 12-1. Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register (IICA) Slave address register (SVA)
Control registers	Peripheral enable register 0 (PER0) IICA control register 0 (IICCTL0) IICA status register (IICS) IICA flag register (IICF) IICA control register 1 (IICCTL1) IICA low-level width setting register (IICWL) IICA high-level width setting register (IICWH) Port mode register 6 (PM6) Port register 6 (P6)

(1) IICA shift register (IICA)

The IICA register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICA register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to the IICA register.

Cancel the wait state and start data transfer by writing data to the IICA register during the wait period.

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

Reset signal generation clears IICA to 00H.

Figure 12-3. Format of IICA Shift Register (IICA)

Address: FFF50H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICA								

Cautions 1. Do not write data to the IICA register during data transfer.

2. Write or read the IICA register only during the wait period. Accessing the IICA register in a communication state other than during the wait period is prohibited. When the device serves as the master, however, the IICA register can be written only once after the communication trigger bit (STT) is set to 1.

3. When communication is reserved, write data to the IICA register after the interrupt triggered by a stop condition is detected.

(2) Slave address register (SVA)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

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

However, rewriting to this register is prohibited while STD = 1 (while the start condition is detected).

Reset signal generation clears the SVA register to 00H.

Figure 12-4. Format of Slave Address Register (SVA)

Address: F0234H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA	A6	A5	A4	A3	A2	A1	A0	0Note

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDA0 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICA) when the address received by this register matches the address value set to the slave address register (SVA) or when an extension code is received.

(5) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(6) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIICA).

An I^2C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by the WTIM bit)
- Interrupt request generated when a stop condition is detected (set by the SPIE bit)

Remark WTIM bit: Bit 3 of IICA control register 0 (IICCTL0)
SPIE bit: Bit 4 of IICA control register 0 (IICCTL0)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCL0 pin from a sampling clock.

(8) Serial clock wait controller

This circuit controls the wait timing.

(9) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate and detect each status.

(10) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(11) Start condition generator

This circuit generates a start condition when the STT bit is set to 1.

However, in the communication reservation disabled status (IICRSV bit = 1), when the bus is not released (IICBSY bit = 1), start condition requests are ignored and the STCF bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPT bit is set to 1.

(13) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCEN bit.

Remark STT bit: Bit 1 of IICA control register 0 (IICCTL0)
SPT bit: Bit 0 of IICA control register 0 (IICCTL0)
IICRSV bit: Bit 0 of IICA flag register (IICF)
IICBSY bit: Bit 6 of IICA flag register (IICF)
STCF bit: Bit 7 of IICA flag register (IICF)
STCEN bit: Bit 1 of IICA flag register (IICF)

12.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following eight registers.

- Peripheral enable register 0 (PER0)
- IICA control register 0 (IICCTL0)
- IICA flag register (IICF)
- IICA status register (IICS)
- IICA control register 1 (IICCTL1)
- IICA low-level width setting register (IICWL)
- IICA high-level width setting register (IICWH)
- 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 IICA is used, be sure to set bit 4 (IICAEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 12-5. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

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

Caution When setting serial interface IICA, be sure to set the IICAEN bit to 1 first. If IICAEN = 0, writing to a control register of serial interface IICA 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) IICA control register 0 (IICCTL0)

This register is used to enable/stop I^2C operations, set wait timing, and set other I^2C operations.

The IICCTL0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE, WTIM, and ACKE bits while IICE = 0 or during the wait period. These bits can be set at the same time when the IICE bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Figure 12-6. Format of IICA Control Register 0 (IICCTL0) (1/4)

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTL0	IICE	LREL	WREL	SPIE	WTIM	ACKE	STT	SPT

IICE	I^2C operation enable
0	Stop operation. Reset the IICA status register (IICS)[Note 1]. 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 (IICE = 0)	Condition for setting (IICE = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL[Notes 2, 3]	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL0 and SDA0 lines are set to high impedance. The following flags of IICA control register 0 (IICCTL0) and the IICA status register (IICS) are cleared to 0. • STT • SPT • MSTS • EXC • COI • TRC • ACKD • STD
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LREL = 0)	Condition for setting (LREL = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL[Notes 2, 3]	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When the WREL bit is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC = 1), the SDA0 line goes into the high impedance state (TRC = 0).	
Condition for clearing (WREL = 0)	Condition for setting (WREL = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICA status register (IICS), the STCF and IICBSY bits of the IICA flag register (IICF), and the CLD and DAD bits of IICA control register 1 (IICCTL1) are reset.

2. The signal of this bit is invalid while IICE0 is 0.

3. When the LREL and WREL bits are read, 0 is always read.

Caution If the operation of I^2C is enabled (IICE = 1) when the SCL0 line is high level, the SDA0 line is low level, and the digital filter is turned on (DFC bit of IICCTL1 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LREL bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICE = 1).

Figure 12-6. Format of IICA Control Register 0 (IICCTL0) (2/4)

SPIE[Note 1]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If the WUP bit of IICA control register 1 (IICCTL1) is 1, no stop condition interrupt will be generated even if SPIE = 1.	
Condition for clearing (SPIE = 0)	Condition for setting (SPIE = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM[Note 1]	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 ($\overline{ACK}$) 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 (WTIM = 0)	Condition for setting (WTIM = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE[Notes 1, 2]	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDA0 line is set to low level.
Condition for clearing (ACKE = 0)	Condition for setting (ACKE = 1)
• Cleared by instruction • Reset	• Set by instruction

Notes 1. The signal of this bit is invalid while IICE0 is 0. Set this bit during that period.

2. The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Figure 12-6. Format of IICA Control Register 0 (IICCTL0) (3/4)

STT[Note]	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSY = 0): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation function is enabled (IICRSV = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSV = 1) Even if this bit is set (1), the STT bit is cleared and the STT clear flag (STCF) is set (1). No start condition is generated. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when the ACKE bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as stop condition trigger (SPT).
- Setting the STT bit to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (STT = 0)	Condition for setting (STT = 1)
• Cleared by setting the STT bit to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL = 1 (exit from communications) • When IICE = 0 (operation stop) • Reset	• Set by instruction

Note The signal of this bit is invalid while IICE0 is 0.

Remarks 1. Bit 1 (STT) becomes 0 when it is read after data setting.

2. IICRSV: Bit 0 of IIC flag register (IICF)
STCF: Bit 7 of IIC flag register (IICF)

Figure 12-6. Format of IICA Control Register 0 (IICCTL0) (4/4)

SPT	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer).

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when the ACKE bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as start condition trigger (STT).
- The SPT bit can be set to 1 only when in master mode.
- When the WTIM bit has been cleared to 0, if the SPT bit 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. The WTIM bit should be changed from 0 to 1 during the wait period following the output of eight clocks, and the SPT bit should be set to 1 during the wait period that follows the output of the ninth clock.
- Setting the SPT bit to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (SPT = 0)	Condition for setting (SPT = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL = 1 (exit from communications) • When IICE = 0 (operation stop) • Reset	• Set by instruction

Caution When bit 3 (TRC) of the IICA status register (IICS) is set to 1 (transmission status), bit 5 (WREL) of IICA control register 0 (IICCTL0) is set to 1 during the ninth clock and wait is canceled, after which the TRC bit is cleared (reception status) and the SDA0 line is set to high impedance. Release the wait performed while the TRC bit is 1 (transmission status) by writing to the IICA shift register.

Remark Bit 0 (SPT) becomes 0 when it is read after data setting.

(3) IICA status register (IICS)

This register indicates the status of I^2C.

The IICS register is read by a 1-bit or 8-bit memory manipulation instruction only when STT = 1 and during the wait period.

Reset signal generation clears this register to 00H.

Caution Reading the IICS register while the address match wakeup function is enabled (WUP = 1) in STOP mode is prohibited. When the WUP bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA interrupt request, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (SPIE = 1) the interrupt generated by detecting a stop condition and read the IICS register after the interrupt has been detected.

Remark STT: bit 1 of IICA control register 0 (IICCTL0)
WUP: bit 7 of IICA control register 1 (IICCTL1)

Figure 12-7. Format of IICA Status Register (IICS) (1/3)

Address: FFF51H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS	MSTS	ALD	EXC	COI	TRC	ACKD	STD	SPD

MSTS	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS = 0)	Condition for setting (MSTS = 1)
• When a stop condition is detected • When ALD = 1 (arbitration loss) • Cleared by LREL = 1 (exit from communications) • When the IICE bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. The MSTS bit is cleared.

Condition for clearing (ALD = 0)	Condition for setting (ALD = 1)
• Automatically cleared after the IICS register is read[Note] • When the IICE bit changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than the IICS register. Therefore, when using the ALD bit, read the data of this bit before the data of the other bits.

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

Figure 12-7. Format of IICA Status Register (IICS) (2/3)

EXC	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC = 0)	Condition for setting (EXC = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 (exit from communications) • When the IICE bit changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

COI	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI = 0)	Condition for setting (COI = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 (exit from communications) • When the IICE bit changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register (SVA)) (set at the rising edge of the eighth clock).

TRC	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 (TRC = 0)	Condition for setting (TRC = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL = 1 (exit from communications) • When the IICE bit changes from 1 to 0 (operation stop) • Cleared by WREL = 1[Note] (wait cancel) • When the ALD bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTS, EXC, COI = 0) <Master> • When “1” is output to the first byte’s LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When “0” is input to the first byte’s LSB (transfer direction specification bit)	<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer) <Slave> • When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)

Note When bit 3 (TRC) of the IICA status register (IICS) is set to 1 (transmission status), bit 5 (WREL) of IICA control register 0 (IICCTL0) is set to 1 during the ninth clock and wait is canceled, after which the TRC bit is cleared (reception status) and the SDA0 line is set to high impedance. Release the wait performed while the TRC bit is 1 (transmission status) by writing to the IICA shift register.

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

Figure 12-7. Format of IICA Status Register (IICS) (3/3)

ACKD	Detection of acknowledge ($\overline{ACK}$)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKD = 0)	Condition for setting (ACKD = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL = 1 (exit from communications) • When the IICE bit changes from 1 to 0 (operation stop) • Reset	• After the SDA0 line is set to low level at the rising edge of SCL0 line's ninth clock

STD	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 (STD = 0)	Condition for setting (STD = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL = 1 (exit from communications) • When the IICE bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD	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 (SPD = 0)	Condition for setting (SPD = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When the IICE bit changes from 1 to 0 (operation stop) <R> • When WUP changes from 1 to 0 • Reset	• When a stop condition is detected

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

(4) IICA flag register (IICF)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.

The IICF register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STT clear flag (STCF) and I^2C bus status flag (IICBSY) bits are read-only.

The IICRSV bit can be used to enable/disable the communication reservation function.

The STCEN bit can be used to set the initial value of the IICBSY bit.

The IICRSV and STCEN bits can be written only when the operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) = 0). When operation is enabled, the IICF register can be read.

Reset signal generation clears this register to 00H.

Figure 12-8. Format of IICA Flag Register (IICF)

Address: FFF52H After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICF	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	STT clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear the STT flag

Condition for clearing (STCF = 0)	Condition for setting (STCF = 1)
• Cleared by STT = 1 • When IICE = 0 (operation stop) • Reset	• Generating start condition unsuccessful and the STT bit cleared to 0 when communication reservation is disabled (IICRSV = 1).

IICBSY	I^2C 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)
• Detection of stop condition • When IICE = 0 (operation stop) • Reset	• Detection of start condition • Setting of the IICE bit when STCEN = 0

STCEN	Initial start enable trigger
0	After operation is enabled (IICE = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN = 0)	Condition for setting (STCEN = 1)
• Cleared by instruction • Detection of start condition • Reset	• Set by instruction

IICRSV	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV = 0)	Condition for setting (IICRSV = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to the STCEN bit only when the operation is stopped (IICE = 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 (STT = 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 (IICE = 0).

Remark STT: Bit 1 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

(5) IICA control register 1 (IICCTL1)

This register is used to set the operation mode of I^2C and detect the statuses of the SCL0 and SDA0 pins.

The IICCTL1 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD and DAD bits are read-only.

Set the IICCTL1 register, except the WUP bit, while operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) is 0).

Reset signal generation clears this register to 00H.

Figure 12-9. Format of IICA Control Register 1 (IICCTL1) (1/2)

Address: F0231H After reset: 00H R/W[Note 1]

Symbol	7	6	<5>	<4>	<3>	<2>	1	0
IICCTL1	WUP	0	CLD	DAD	SMC	DFC	0	0

WUP	Control of address match wakeup
0	Stops operation of address match wakeup function in STOP mode.
1	Enables operation of address match wakeup function in STOP mode.

To shift to STOP mode when WUP = 1, execute the STOP instruction at least three clocks after setting (1) the WUP bit (see **Figure 12-22 Flow When Setting WUP = 1**).

Clear (0) the WUP bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUP bit. (The wait must be released and transmit data must be written after the WUP bit has been cleared (0).)

The interrupt timing when the address has matched or when an extension code has been received, while WUP = 1, is identical to the interrupt timing when WUP = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUP = 1, a stop condition interrupt is not generated even if the SPIE bit is set to 1.

When WUP = 0 is set by a source other than an interrupt from serial interface IICA, operation as the master device cannot be performed until the subsequent start condition or stop condition is detected. Do not output a start condition by setting (1) the STT bit, without waiting for the detection of the subsequent start condition or stop condition.

Condition for clearing (WUP = 0)	Condition for setting (WUP = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when the MSTS, EXC, and COI bits are “0”, and the STD bit also “0” (communication not entered))[Note 2]

Notes 1. Bits 4 and 5 are read-only.

2. The status of the IICA status register (IICS) must be checked and the WUP bit must be set during the period shown below.

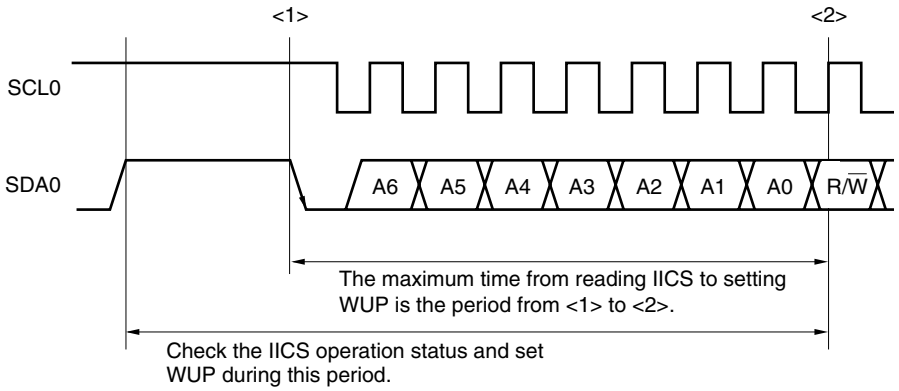


Figure 12-9. Format of IICA Control Register 1 (IICCTL1) (2/2)

CLD	Detection of SCL0 pin level (valid only when IICE = 1)
0	The SCL0 pin was detected at low level.
1	The SCL0 pin was detected at high level.
Condition for clearing (CLD = 0)	Condition for setting (CLD = 1)
• When the SCL0 pin is at low level • When IICE = 0 (operation stop) • Reset	• When the SCL0 pin is at high level

DAD	Detection of SDA0 pin level (valid only when IICE = 1)
0	The SDA0 pin was detected at low level.
1	The SDA0 pin was detected at high level.
Condition for clearing (DAD = 0)	Condition for setting (DAD = 1)
• When the SDA0 pin is at low level • When IICE = 0 (operation stop) • Reset	• When the SDA0 pin is at high level

SMC	Operation mode switching
0	Operates in standard mode.
1	Operates in fast mode.

DFC	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 the DFC bit being set (1) or cleared (0). The digital filter is used for noise elimination in fast mode.	

Remark IICE: Bit 7 of IICA control register 0 (IICCTL0)

(6) IICA low-level width setting register (IICWL)

This register is used to set the low-level width of the SCL0 pin signal that is output by serial interface IICA.

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

Set the IICWL register while operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) is 0).

Reset signal generation sets this register to FFH.

Figure 12-10. Format of IICA Low-Level Width Setting Register (IICWL)

Address: F0232H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWL								

(7) IICA high-level width setting register (IICWH)

This register is used to set the high-level width of the SCL0 pin signal that is output by serial interface IICA.

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

Set the IICWL register while operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) is 0).

Reset signal generation sets this register to FFH.

Figure 12-11. Format of IICA High-Level Width Setting Register (IICWH)

Address: F0233H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWH								

Remark For how to set the transfer clock by using the IICWL and IICWH registers, see **12.4.2 Setting transfer clock by using IICWL and IICWH registers**.

(8) 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 the IICE bit (bit 7 of IICA control register 0 (IICCTL0)) to 1 before setting the output mode because the P60/SCL0 and P61/SDA0 pins output a low level (fixed) when the IICE bit is 0.

The PM6 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 12-12. Format of Port Mode Register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

12.4 I^2C Bus Mode Functions

12.4.1 Pin configuration

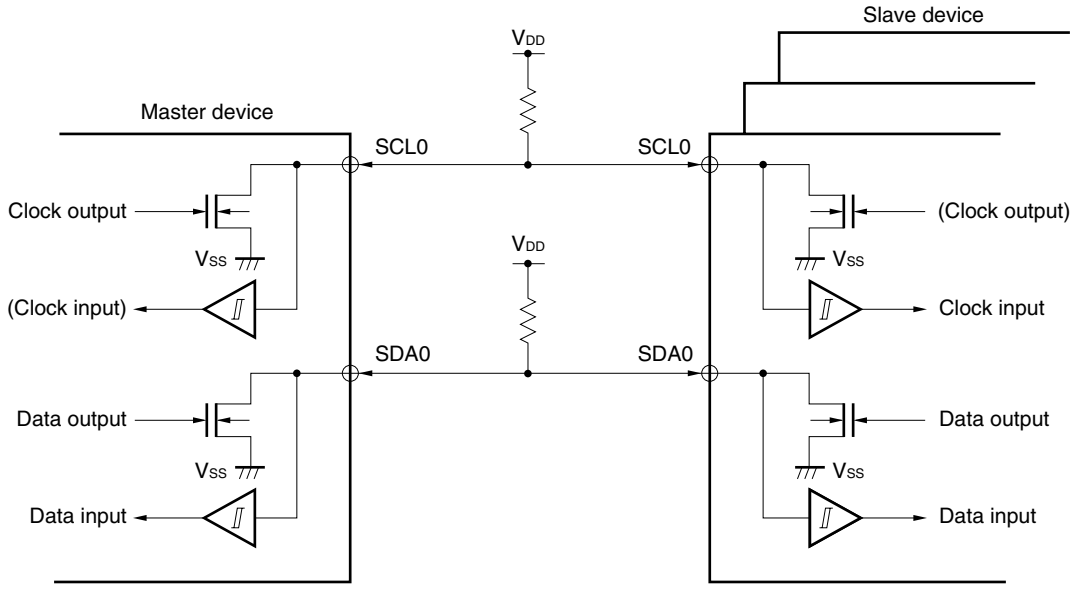
The serial clock pin (SCL0) and the serial data bus pin (SDA0) are configured as follows.

(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 12-13. Pin Configuration Diagram



12.4.2 Setting transfer clock by using IICWL and IICWH registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{CLK}}{IICWL + IICWH + f_{CLK}(t_R + t_F)}$$

At this time, the optimal setting values of the IICWL and IICWH registers are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICWL = \frac{0.52}{\text{Transfer clock}} \times f_{CLK}$$
$$IICWH = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{CLK}$$

- When the standard mode

$$IICWL = \frac{0.47}{\text{Transfer clock}} \times f_{CLK}$$
$$IICWH = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{CLK}$$

(2) Setting IICWL and IICWH registers on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$IICWL = 1.3\ \mu s \times f_{CLK}$$
$$IICWH = (1.2\ \mu s - t_R - t_F) \times f_{CLK}$$

- When the standard mode

$$IICWL = 4.7\ \mu s \times f_{CLK}$$
$$IICWH = (5.3\ \mu s - t_R - t_F) \times f_{CLK}$$

Caution Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.

Fast mode: f_{CLK} = 3.5 MHz (MIN.)

Standard mode: f_{CLK} = 1 MHz (MIN.)

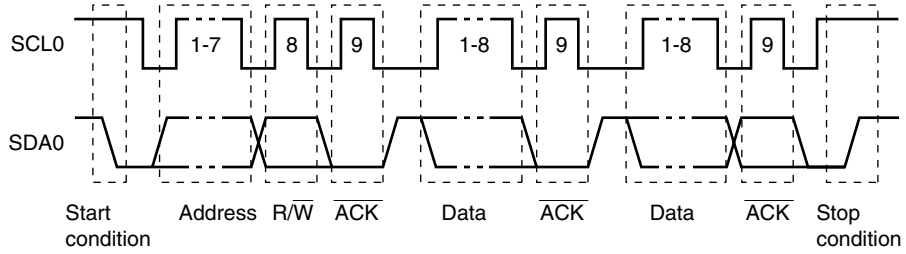
Remarks 1. Calculate the rise time (t_R) and fall time (t_F) of the SDA0 and SCL0 signals separately, because they differ depending on the pull-up resistance and wire load.

2. IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
t_R: SDA0 and SCL0 signal rising times
f_{CLK}: CPU/peripheral hardware clock frequency

12.5 I²C Bus Definitions and Control Methods

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus. Figure 12-14 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 12-14. I²C Bus Serial Data Transfer Timing



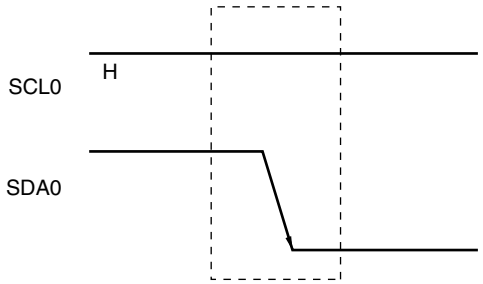
The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCL0) is continuously output by the master device. However, in the slave device, the SCL0 pin low level period can be extended and a wait can be inserted.

12.5.1 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 12-15. Start Conditions



A start condition is output when bit 1 (STT) of IICA control register 0 (IICCTL0) is set (1) after a stop condition has been detected (SPD: Bit 0 of the IICA status register (IICS) = 1). When a start condition is detected, bit 1 (STD) of the IICS register is set (1).

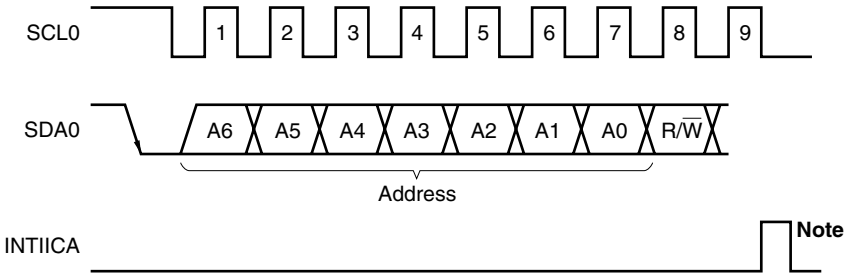
12.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register (SVA). If the address data matches the SVA register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 12-16. Address



Note INTIICA is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **12.5.3 Transfer direction specification** are written to the IICA shift register (IICA). The received addresses are written to the IICA register.

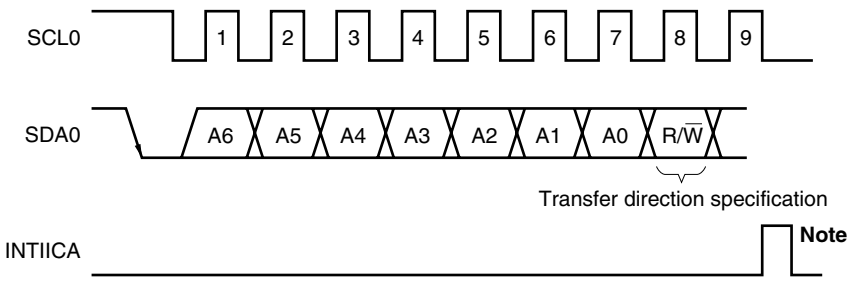
The slave address is assigned to the higher 7 bits of the IICA register.

12.5.3 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 12-17. Transfer Direction Specification



Note INTIICA is not issued if data other than a local address or extension code is received during slave device operation.

12.5.4 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 (ACKD) of the IICA status register (IICS).

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.

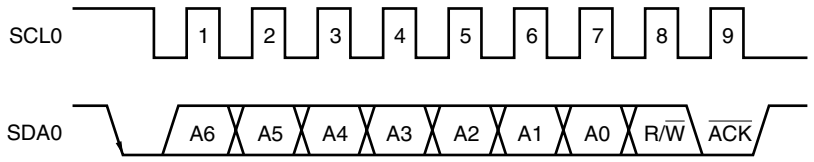
To generate $\overline{ACK}$, the reception side makes the SDA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{ACK}$ is enabled by setting bit 2 (ACKE) of IICA control register 0 (IICCTL0) to 1. Bit 3 (TRC) of the IICS register is set by the data of the eighth bit that follows 7-bit address information. Usually, set the ACKE bit to 1 for reception (TRC = 0).

If a slave can receive no more data during reception (TRC = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKE bit to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC = 0), it must clear the ACKE bit 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 12-18. $\overline{ACK}$



When the local address is received, $\overline{ACK}$ is automatically generated, regardless of the value of the ACKE bit. 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 the ACKE bit 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.

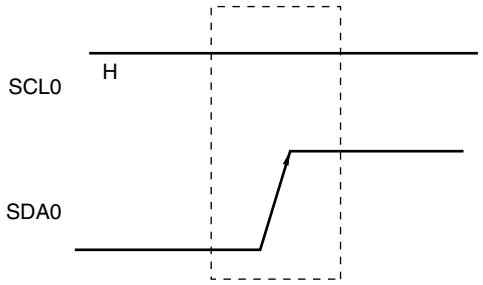
- When 8-clock wait state is selected (bit 3 (WTIM) of IICCTL0 register = 0):
  By setting the ACKE bit to 1 before releasing the wait state, $\overline{ACK}$ is generated at the falling edge of the eighth clock of the SCL0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM) of IICCTL0 register = 1):
  $\overline{ACK}$ is generated by setting the ACKE bit to 1 in advance.

12.5.5 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 12-19. Stop Condition



A stop condition is generated when bit 0 (SPT) of IICA control register 0 (IICCTL0) is set to 1. When the stop condition is detected, bit 0 (SPD) of the IICA status register (IICS) is set to 1 and INTIICA is generated when bit 4 (SPIE) of the IICCTL0 register is set to 1.

12.5.6 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 12-20. 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 ACKE = 1)

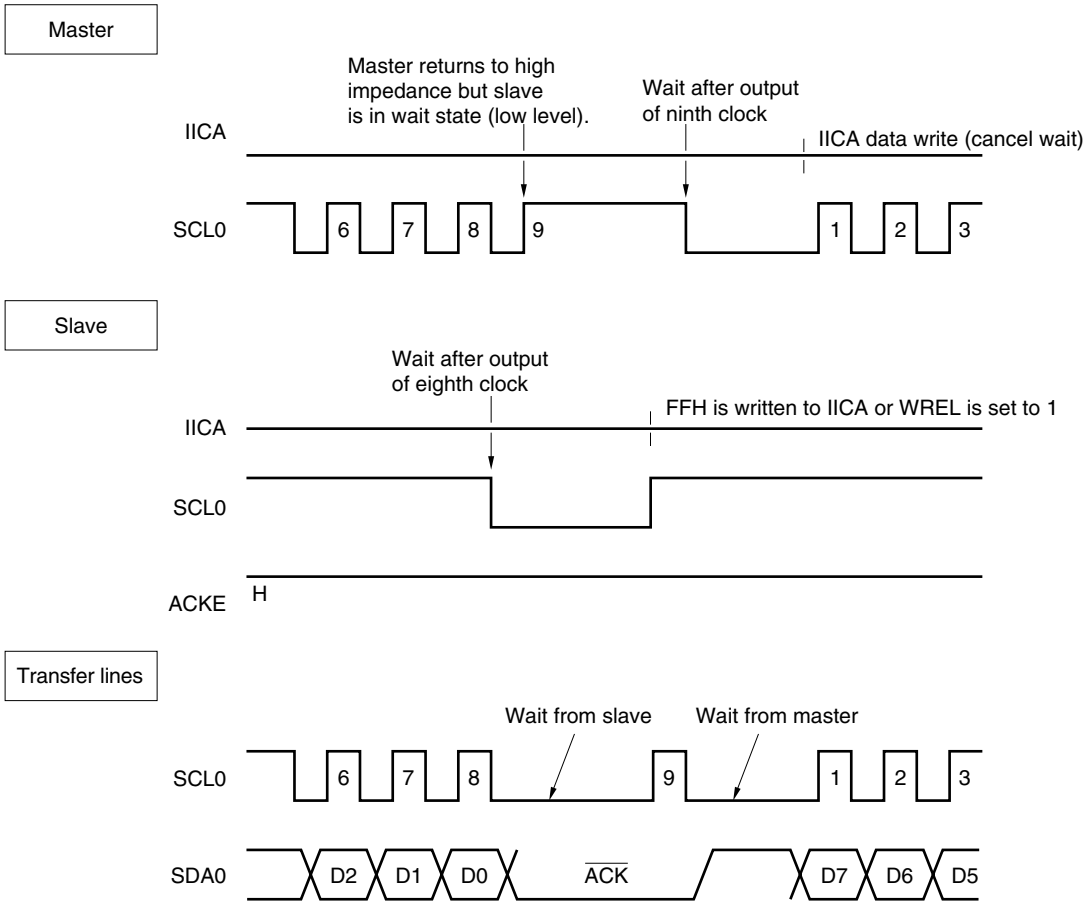
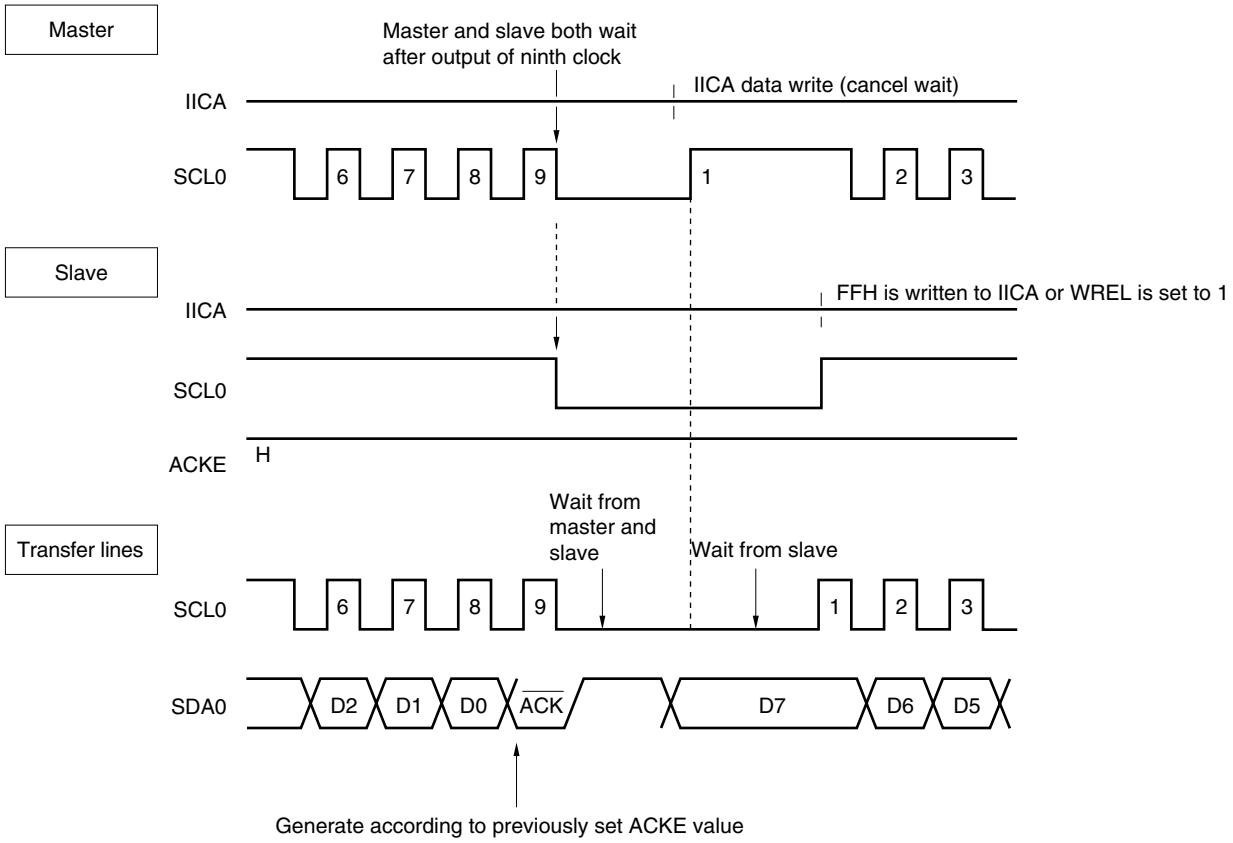


Figure 12-20. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE = 1)



Remark ACKE: Bit 2 of IICA control register 0 (IICCTL0)
WREL: Bit 5 of IICA control register 0 (IICCTL0)

A wait may be automatically generated depending on the setting of bit 3 (WTIM) of IICA control register 0 (IICCTL0).

Normally, the receiving side cancels the wait state when bit 5 (WREL) of the IICCTL0 register is set to 1 or when FFH is written to the IICA shift register (IICA), and the transmitting side cancels the wait state when data is written to the IICA register.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT) of the IICCTL0 register to 1
- By setting bit 0 (SPT) of the IICCTL0 register to 1

12.5.7 Canceling wait

The I^2C usually cancels a wait state by the following processing.

- Writing data to the IICA shift register (IICA)
- Setting bit 5 (WREL) of IICA control register 0 (IICCTL0) (canceling wait)
- Setting bit 1 (STT) of the IICCTL0 register (generating start condition)[Note]
- Setting bit 0 (SPT) of the IICCTL0 register (generating stop condition)[Note]

Note Master only

When the above wait canceling processing is executed, the I^2C cancels the wait state and communication is resumed.

To cancel a wait state and transmit data (including addresses), write the data to the IICA register.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL) of the IICCTL0 register to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT) of the IICCTL0 register to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT) of the IICCTL0 register to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to the IICA register after canceling a wait state by setting the WREL bit to 1, an incorrect value may be output to SDA0 line because the timing for changing the SDA0 line conflicts with the timing for writing the IICA register.

In addition to the above, communication is stopped if the IICE bit is cleared to 0 when communication has been aborted, so that the wait state can be canceled.

If the I^2C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LREL) of the IICCTL0 register, so that the wait state can be canceled.

Caution If a processing to cancel a wait state is executed when WUP = 1, the wait state will not be canceled.

12.5.8 Interrupt request (INTIICA) generation timing and wait control

The setting of bit 3 (WTIM) of IICA control register 0 (IICCTL0) determines the timing by which INTIICA is generated and the corresponding wait control, as shown in Table 12-2.

Table 12-2. INTIICA Generation Timing and Wait Control

WTIM	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9 Notes 1, 2	8 Note 2	8 Note 2	9	8	8
1	9 Notes 1, 2	9 Note 2	9 Note 2	9	9	9

Notes 1. The slave device's INTIICA signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to the slave address register (SVA).
At this point, $\overline{ACK}$ is generated regardless of the value set to the IICCTL0 register's bit 2 (ACKE). For a slave device that has received an extension code, INTIICA occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICA 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 the slave address register (SVA) and extension code is not received, neither INTIICA 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 WTIM bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to the IICA shift register (IICA)
- Setting bit 5 (WREL) of IICA control register 0 (IICCTL0) (canceling wait)
- Setting bit 1 (STT) of IICCTL0 register (generating start condition) Note
- Setting bit 0 (SPT) of IICCTL0 register (generating stop condition) Note

Note Master only.

When an 8-clock wait has been selected (WTIM = 0), the presence/absence of $\overline{ACK}$ generation must be determined prior to wait cancellation.

(5) Stop condition detection

INTIICA is generated when a stop condition is detected (only when SPIE = 1).

12.5.9 Address match detection method

In I^2C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt request (INTIICA) occurs when the address set to the slave address register (SVA) matches the slave address sent by the master device, or when an extension code has been received.

12.5.10 Error detection

In I^2C bus mode, the status of the serial data bus (SDA0) during data transmission is captured by the IICA shift register (IICA) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

12.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC) is set to 1 for extension code reception and an interrupt request (INTIICA) is issued at the falling edge of the eighth clock. The local address stored in the slave address register (SVA) is not affected.

(2) The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer when the SVA register is set to 11110xx0. Note that INTIICA occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC = 1
- Seven bits of data match: COI = 1

Remark EXC: Bit 5 of IICA status register (IICS)
COI: Bit 4 of IICA status register (IICS)

(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 (LREL) of IICA control register 0 (IICCTL0) to 1 to set the standby mode for the next communication operation.

Table 12-3. Bit Definitions of Major Extension Codes

Slave Address	R/W Bit	Description
0 0 0 0 0 0 0	0	General call address
1 1 1 1 0 x x	0	10-bit slave address specification (during address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (after address match, when read command is issued)

Remark See the I^2C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

12.5.12 Arbitration

When several master devices simultaneously generate a start condition (when the STT bit is set to 1 before the STD bit is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (ALD) in the IICA status register (IICS) 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 ALD = 1 setting that has been made by software.

For details of interrupt request timing, see **12.5.8 Interrupt request (INTIICA) generation timing and wait control**.

Remark STD: Bit 1 of IICA status register (IICS)
STT: Bit 1 of IICA control register 0 (IICCTL0)

Figure 12-21. Arbitration Timing Example

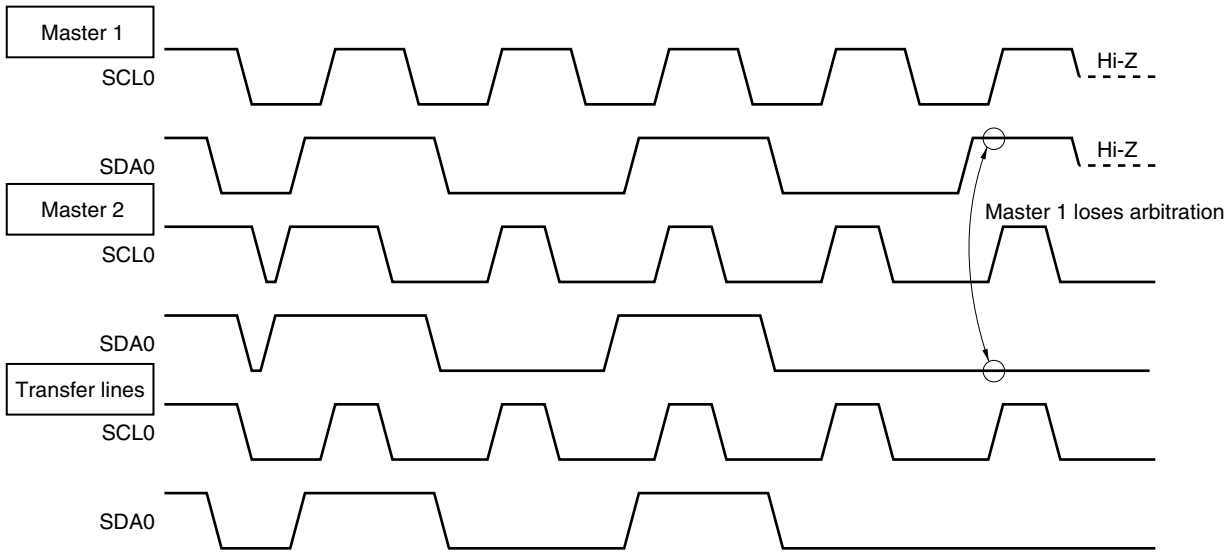


Table 12-4. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ 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 SPIE = 1)[Note 2]
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIE = 1)[Note 2]
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When SCL0 is at low level while attempting to generate a restart condition	

Notes 1. When the WTIM bit (bit 3 of IICA control register 0 (IICCTL0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM = 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 SPIE = 1 for master device operation.

Remark SPIE: Bit 4 of IICA control register 0 (IICCTL0)

12.5.13 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIICA) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIICA 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 (SPIE) of IICA control register 0 (IICCTL0) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

To use the wakeup function in the STOP mode, set the WUP bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICA) is also generated when a local address and extension code have been received. Operation returns to normal operation by using an instruction to clear (0) the WUP bit after this interrupt has been generated.

Figure 12-22 shows the flow for setting WUP = 1 and Figure 12-23 shows the flow for setting WUP = 0 upon an address match.

Figure 12-22. Flow When Setting WUP = 1

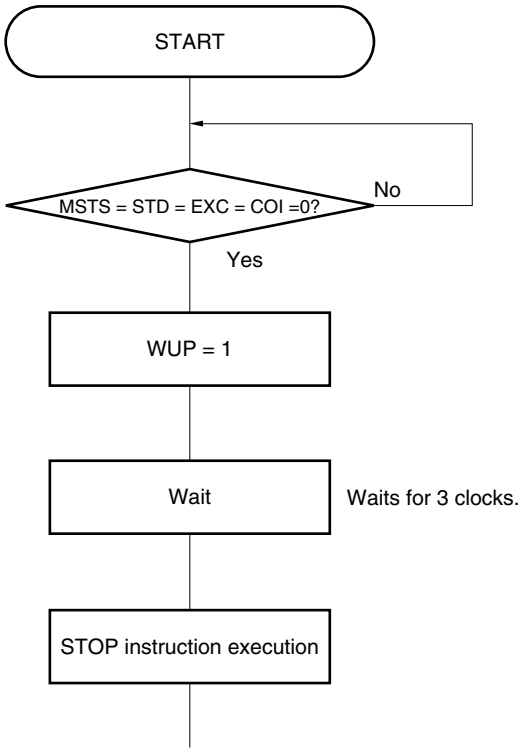
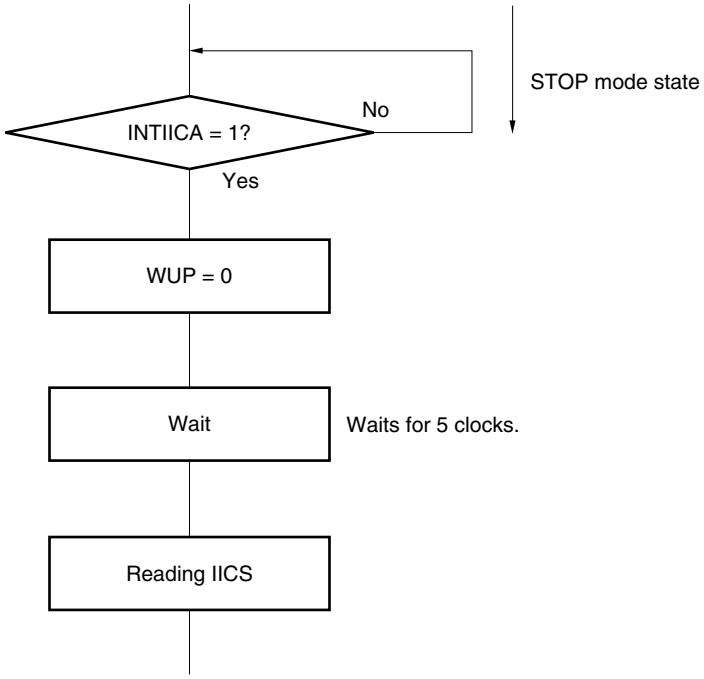


Figure 12-23. Flow When Setting WUP = 0 upon Address Match (Including Extension Code Reception)

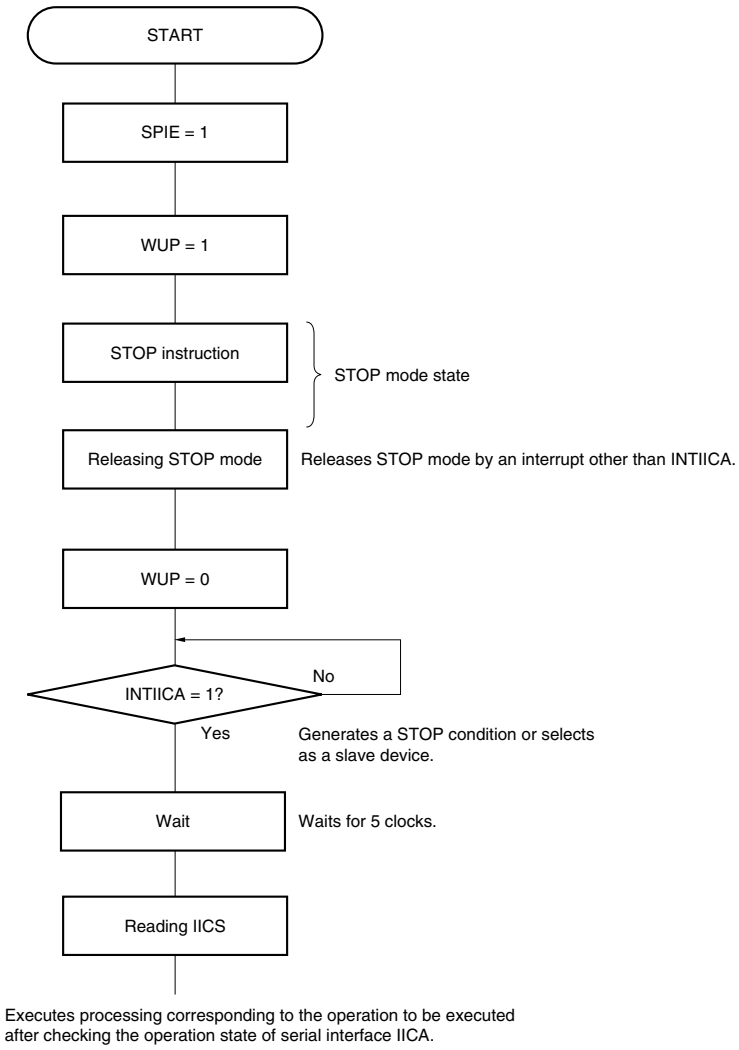


Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICA) generated from serial interface IICA.

- Master device operation: Flow shown in Figure 12-24
- Slave device operation: Same as the flow in Figure 12-23

Figure 12-24. When Operating as Master Device after Releasing STOP Mode other than by INTIICA



12.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IICA flag register (IICF) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL) of IICA control register 0 (IICCTL0) to 1 and saving communication).

If bit 1 (STT) of the IICCTL0 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.
If an address is written to the IICA shift register (IICA) after bit 4 (SPIE) of the IICCTL0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICA) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICA register before the stop condition is detected is invalid.
When the STT bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released ..... a start condition is generated
- If the bus has not been released (standby mode)..... communication reservation

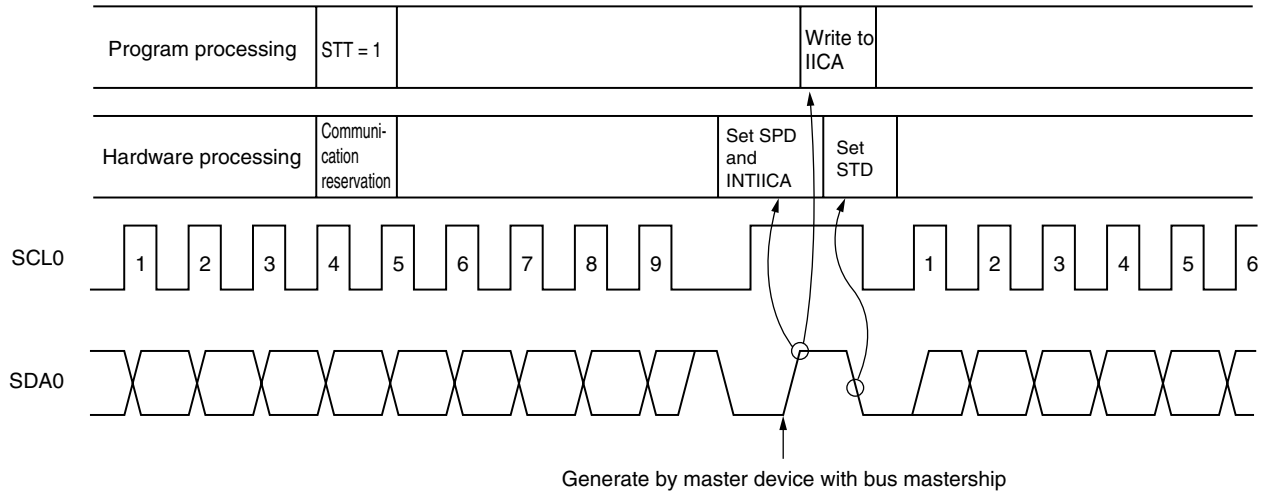
Check whether the communication reservation operates or not by using the MSTS bit (bit 7 of the IICA status register (IICS)) after the STT bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

Wait time from setting STT = 1 to checking the MSTS flag: (IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{CLK}$ [clocks]

Remark IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

Figure 12-25 shows the communication reservation timing.

Figure 12-25. Communication Reservation Timing



Remark IICA: IICA shift register
STT: Bit 1 of IICA control register 0 (IICCTL0)
STD: Bit 1 of IICA status register (IICS)
SPD: Bit 0 of IICA status register (IICS)

Communication reservations are accepted via the timing shown in Figure 12-26. After bit 1 (STD) of the IICA status register (IICS) is set to 1, a communication reservation can be made by setting bit 1 (STT) of IICA control register 0 (IICCTL0) to 1 before a stop condition is detected.

Figure 12-26. Timing for Accepting Communication Reservations

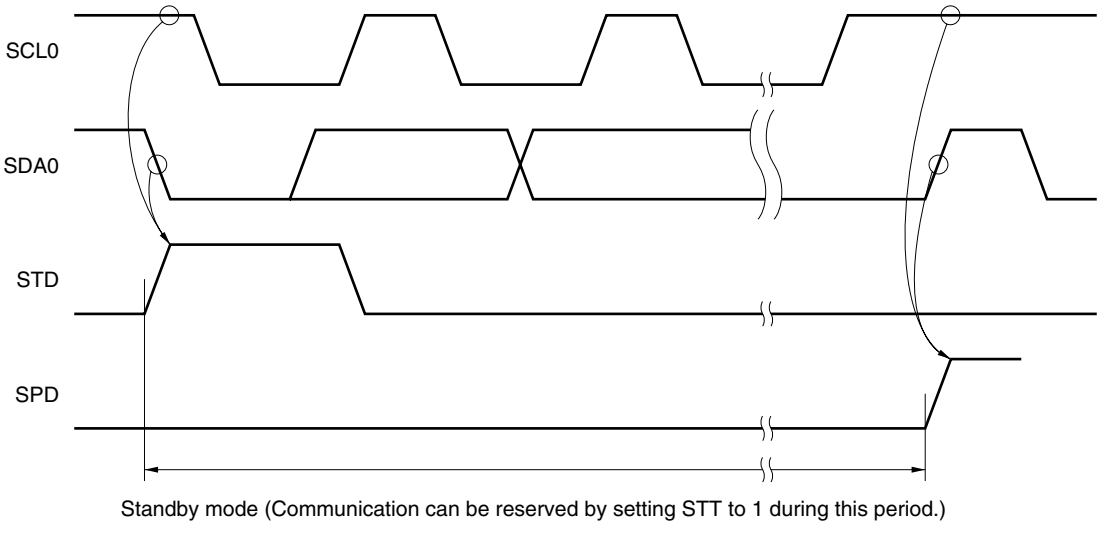
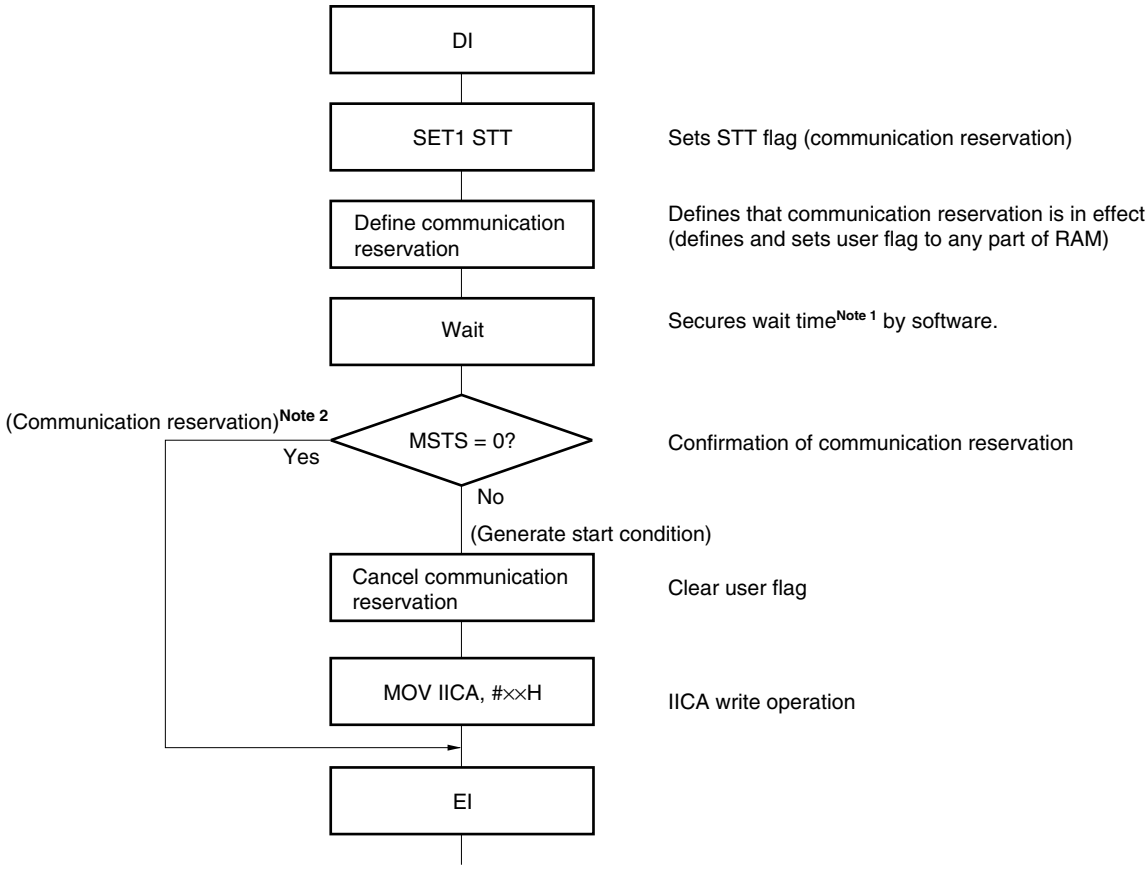


Figure 12-27 shows the communication reservation protocol.

Figure 12-27. Communication Reservation Protocol



Notes 1. The wait time is calculated as follows.
(IICWL setting value + IICWH setting value + 4) + $t_F \times 2 \times f_{CLK}$ [clocks]

2. The communication reservation operation executes a write to the IICA shift register (IICA) when a stop condition interrupt request occurs.

Remark STT: Bit 1 of IICA control register 0 (IICCTL0)
MSTS: Bit 7 of IICA status register (IICS)
IICA: IICA shift register
IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

(2) When communication reservation function is disabled (bit 0 (IICRSV) of IICA flag register (IICF) = 1)

When bit 1 (STT) of IICA control register 0 (IICCTL0) 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 (LREL) of the IICCTL0 register to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCF (bit 7 of the IICF register). It takes up to 5 clocks until the STCF bit is set to 1 after setting STT = 1. Therefore, secure the time by software.

12.5.15 Cautions

(1) When STCEN = 0

Immediately after I^2C operation is enabled (IICE = 1), the bus communication status (IICBSY = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

<1> Set IICA control register 1 (IICCTL1).
<2> Set bit 7 (IICE) of IICA control register 0 (IICCTL0) to 1.
<3> Set bit 0 (SPT) of the IICCTL0 register to 1.

(2) When STCEN = 1

Immediately after I^2C operation is enabled (IICE = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I^2C communications are already in progress

If I^2C operation is enabled and the device participates in communication already in progress when the SDA0 pin is low and the SCL0 pin is high, the macro of I^2C 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^2C communications. To avoid this, start I^2C in the following sequence.

<1> Clear bit 4 (SPIE) of the IICCTL0 register to 0 to disable generation of an interrupt request signal (INTIICA) when the stop condition is detected.
<2> Set bit 7 (IICE) of the IICCTL0 register to 1 to enable the operation of I^2C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL) of the IICCTL0 register to 1 before $\overline{ACK}$ is returned (4 to 80 clocks after setting the IICE bit to 1), to forcibly disable detection.

(4) Setting the STT and SPT bits (bits 1 and 0 of the IICCTL0 register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set the SPIE bit (bit 4 of the IICTL0 register) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to the IICA shift register (IICA) after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set the SPIE bit to 1 when the MSTS bit (bit 7 of the IICA status register (IICS)) is detected by software.

12.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the 78K0R/Kx3-L as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multimaster system

In the I^2C bus multimaster system, whether the bus is released or used cannot be judged by the I^2C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the 78K0R/Kx3-L 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/Kx3-L looses in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

(3) Slave operation

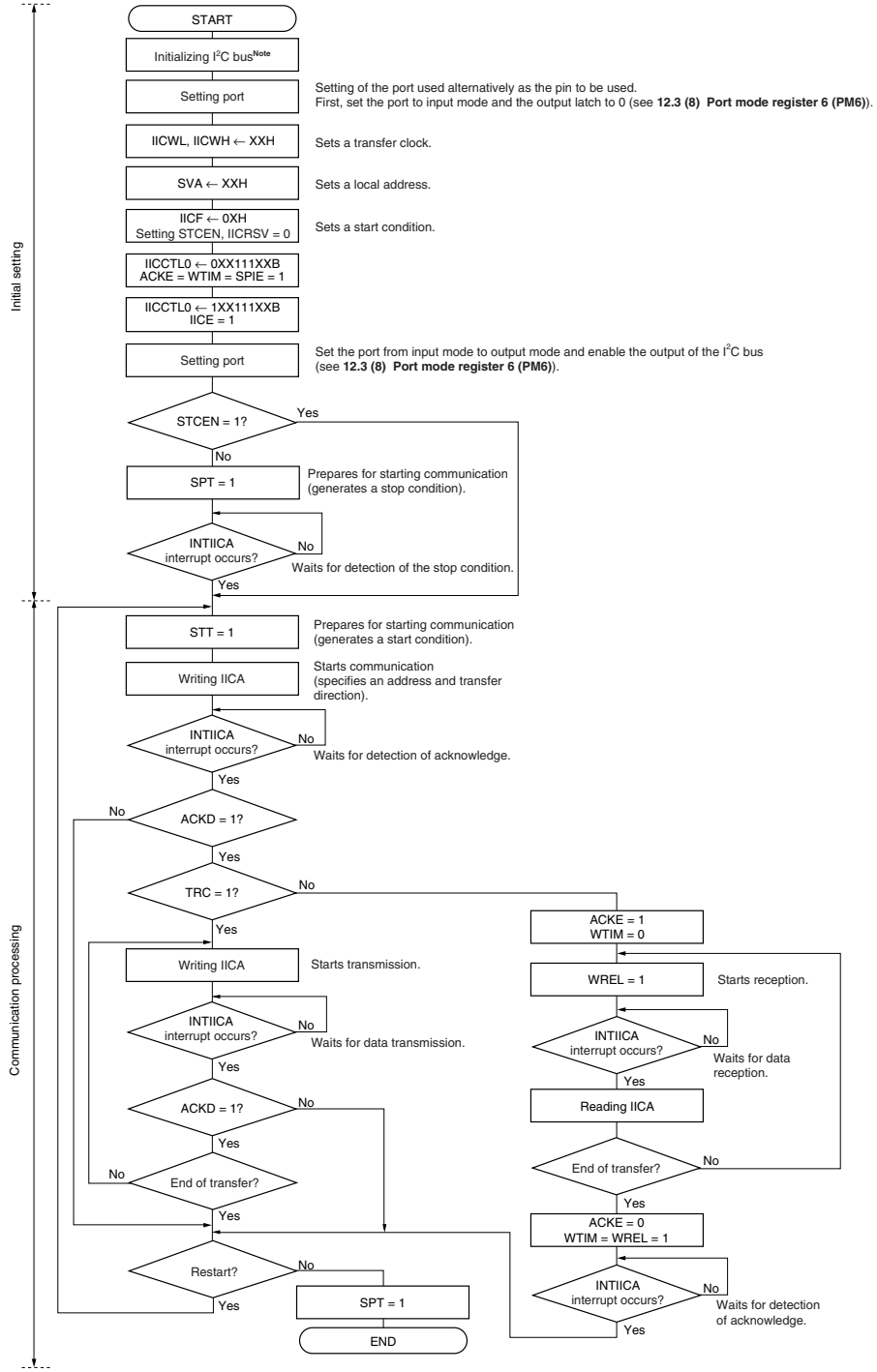
An example of when the 78K0R/Kx3-L is used as the I^2C bus slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICA interrupt occurrence (communication waiting). When an INTIICA 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 12-28. Master Operation in Single-Master System

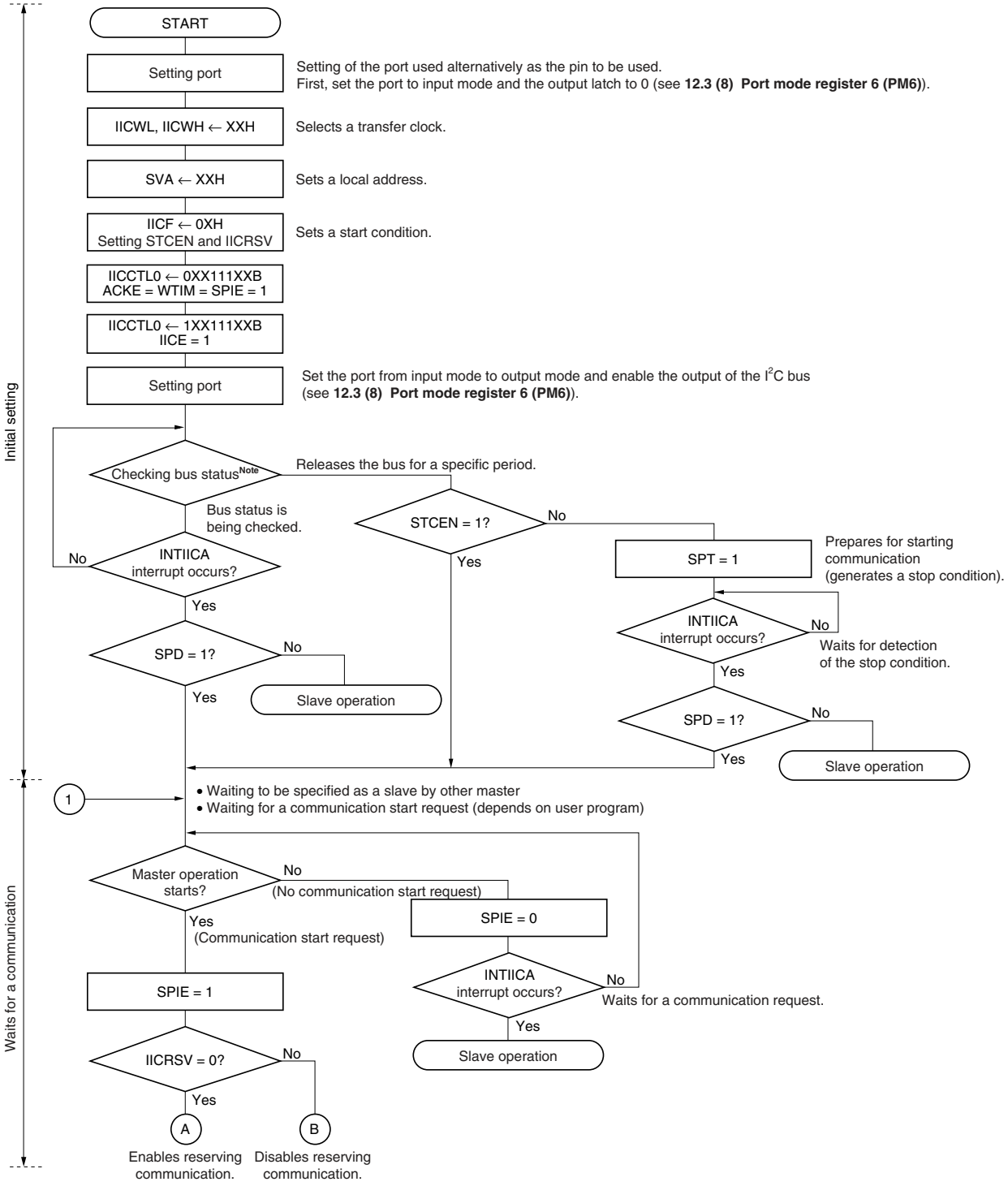


Note Release (SCL0 and SDA0 pins = high level) the I²C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDA0 pin, for example, set the SCL0 pin in the output port mode, and output a clock pulse from the output port until the SDA0 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.

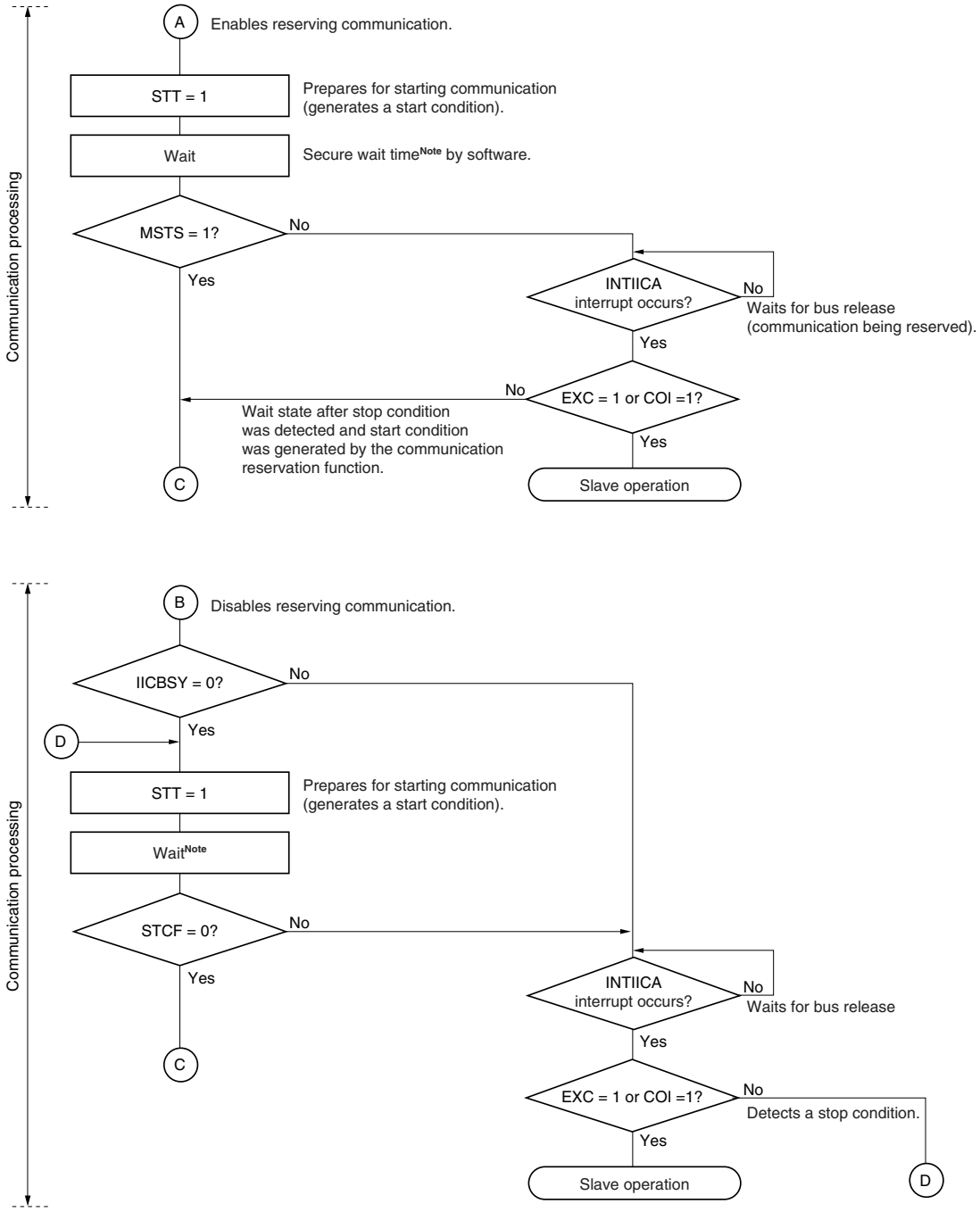
(2) Master operation in multi-master system

Figure 12-29. Master Operation in Multi-Master System (1/3)



Note Confirm that the bus is released (CLD bit = 1, DAD 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^2C bus (SCL0 and SDA0 pins = high level) in conformance with the specifications of the product that is communicating.

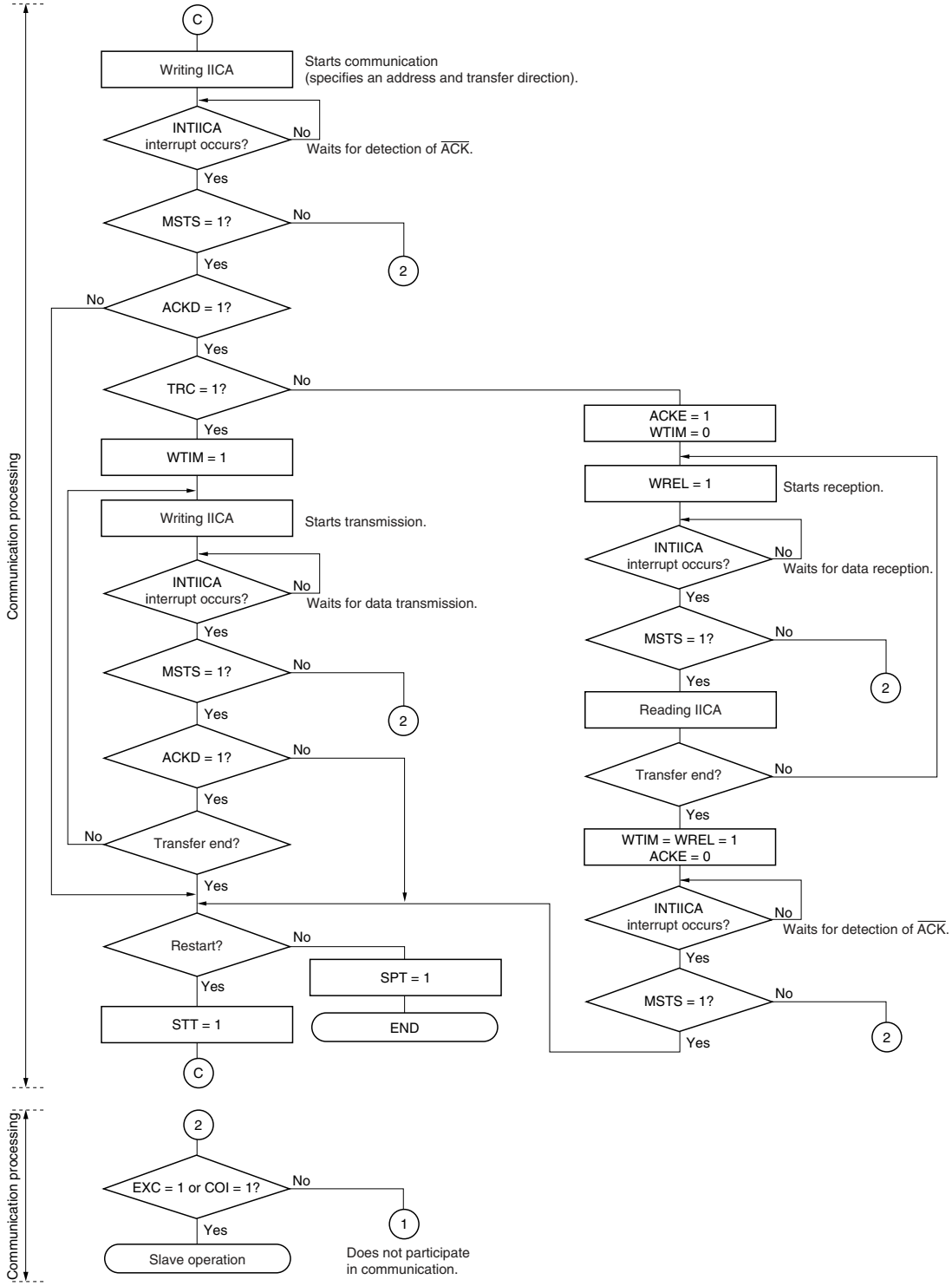
Figure 12-29. Master Operation in Multi-Master System (2/3)



Note The wait time is calculated as follows.
(IICWL setting value + IICWH setting value + 4) × f_{CLK} + t_F × 2 [clocks]

Remark IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

Figure 12-29. 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 MSTS bit each time interrupt INTIICA has occurred to check the arbitration result.

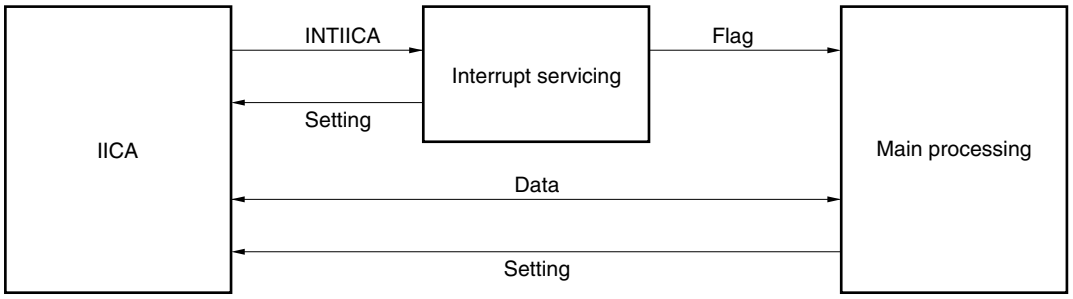
3. To use the device as a slave in a multi-master system, check the status by using the IICA status register (IICS) and IICA flag register (IICF) each time interrupt INTIICA has occurred, and determine the processing to be performed next.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICA 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 INTIICA 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 INTIICA.

<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 $\overline{\text{ACK}}$ from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICA interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as the TRC bit.

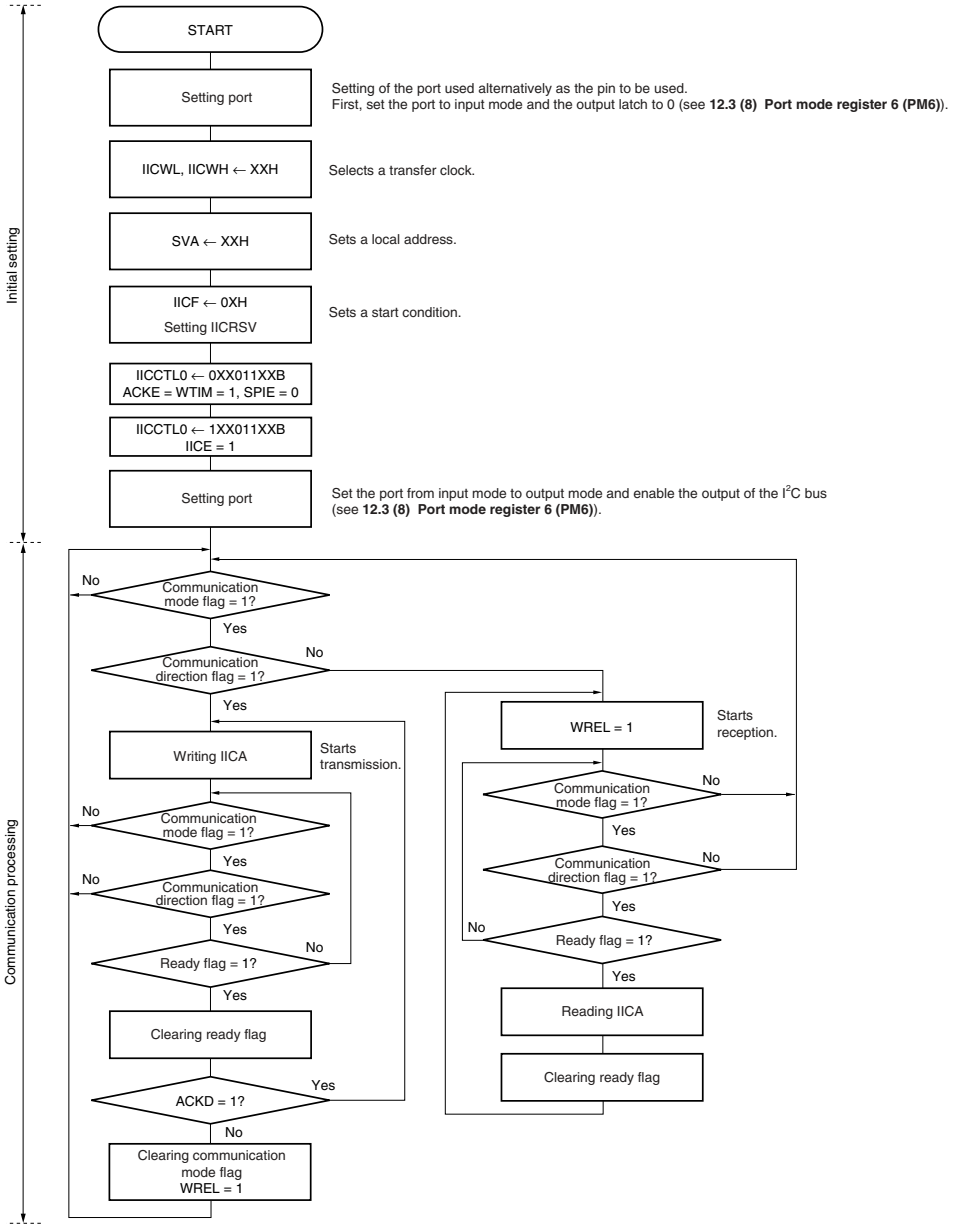
The main processing of the slave operation is explained next.

Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns $\overline{ACK}$. If $\overline{ACK}$ is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, $\overline{ACK}$ 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 12-30. Slave Operation Flowchart (1)



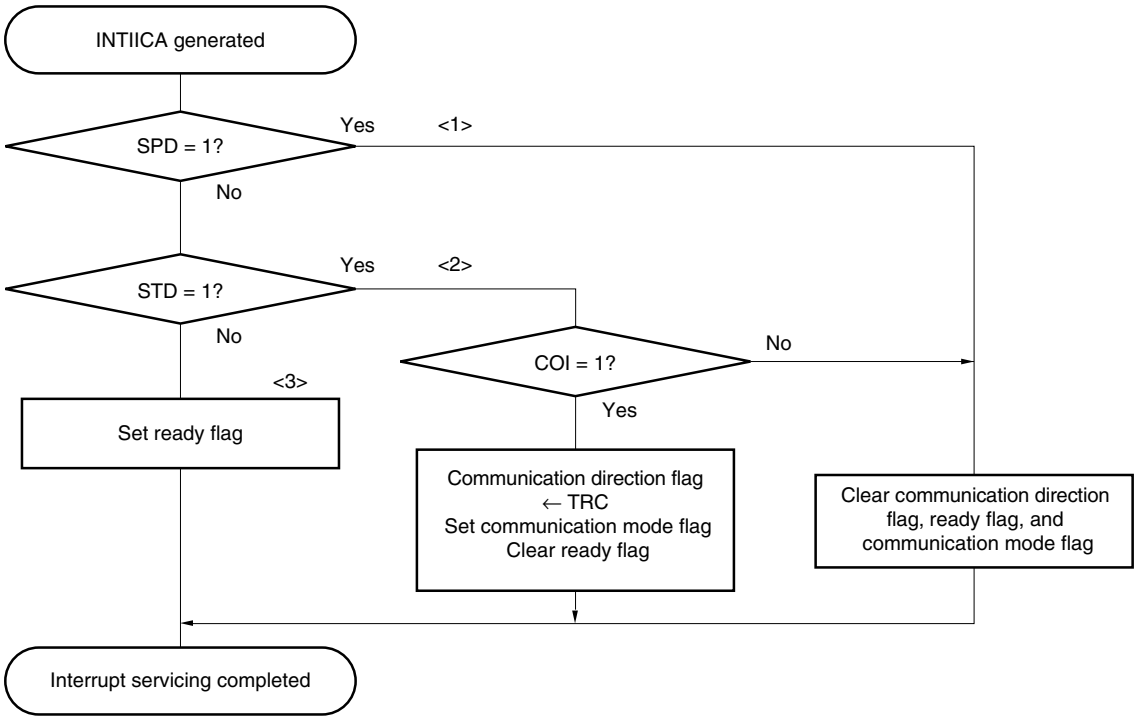
Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

An example of the processing procedure of the slave with the INTIICA interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICA interrupt checks the status, and the following operations are performed.

<1> Communication is stopped if the stop condition is issued.
<2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).
<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in Figure 12-31 Slave Operation Flowchart (2).

Figure 12-31. Slave Operation Flowchart (2)



12.5.17 Timing of I²C interrupt request (INTIICA) occurrence

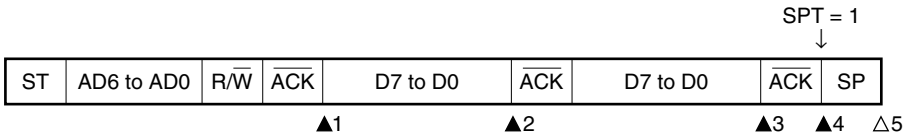
The timing of transmitting or receiving data and generation of interrupt request signal INTIICA, and the value of the IICA status register (IICS) when the INTIICA signal is generated are shown below.

Remark ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM = 0

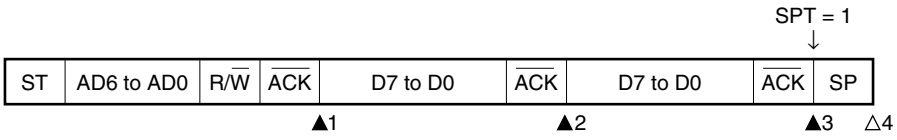


▲1: IICS = 1000×110B
▲2: IICS = 1000×000B
▲3: IICS = 1000×000B (Sets the WTIM bit to 1)[Note]
▲4: IICS = 1000××00B (Sets the SPT bit to 1)[Note]
△5: IICS = 00000001B

Note To generate a stop condition, set the WTIM bit to 1 and change the timing for generating the INTIICA interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

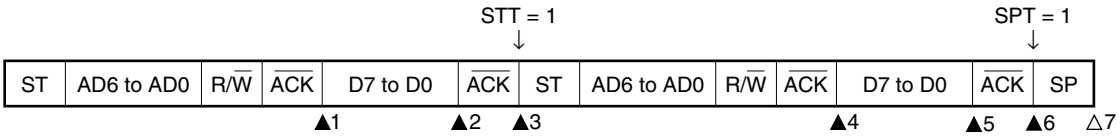


▲1: IICS = 1000×110B
▲2: IICS = 1000×100B
▲3: IICS = 1000××00B (Sets the SPT bit to 1)
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM = 0

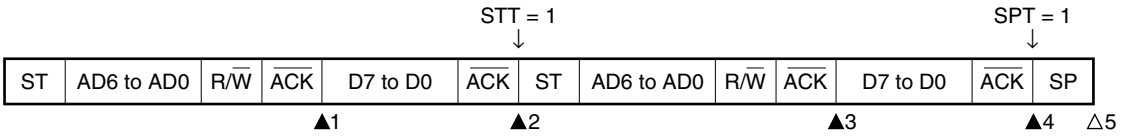


▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets the WTIM bit to 1)[Note 1]
▲3: IICS = 1000××00B (Clears the WTIM bit to 0[Note 2], sets the STT bit to 1)
▲4: IICS = 1000×110B
▲5: IICS = 1000×000B (Sets the WTIM bit to 1)[Note 3]
▲6: IICS = 1000××00B (Sets the SPT bit to 1)
△7: IICS = 00000001B

Notes 1. To generate a start condition, set the WTIM bit to 1 and change the timing for generating the INTIICA interrupt request signal.
2. Clear the WTIM bit to 0 to restore the original setting.
3. To generate a stop condition, set the WTIM bit to 1 and change the timing for generating the INTIICA interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

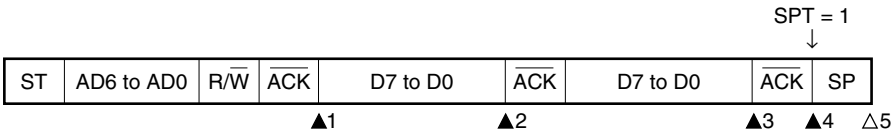


▲1: IICS = 1000×110B
▲2: IICS = 1000××00B (Sets the STT bit to 1)
▲3: IICS = 1000×110B
▲4: IICS = 1000××00B (Sets the SPT bit to 1)
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM = 0

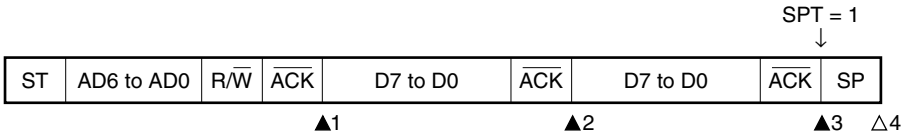


▲1: IICS = 1010×110B
▲2: IICS = 1010×000B
▲3: IICS = 1010×000B (Sets the WTIM bit to 1)[Note]
▲4: IICS = 1010××00B (Sets the SPT bit to 1)
△5: IICS = 00000001B

Note To generate a stop condition, set the WTIM bit to 1 and change the timing for generating the INTIICA interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



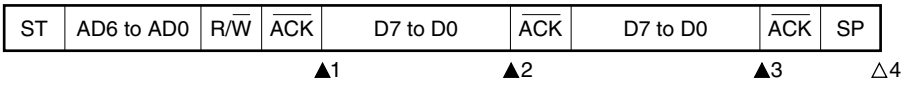
▲1: IICS = 1010×110B
▲2: IICS = 1010×100B
▲3: IICS = 1010××00B (Sets the SPT bit to 1)
△4: IICS = 00001001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

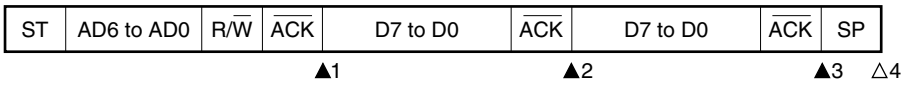
(i) When WTIM = 0



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

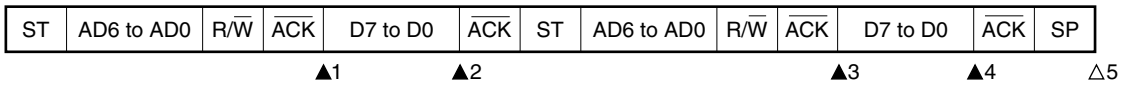


▲1: IICS = 0001×110B
▲2: IICS = 0001×100B
▲3: IICS = 0001××00B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

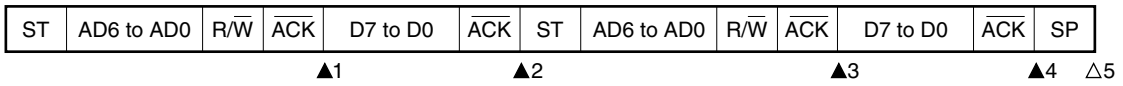
(i) When WTIM = 0 (after restart, matches with SVA)



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×110B
▲4: IICS = 0001×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, matches with SVA)

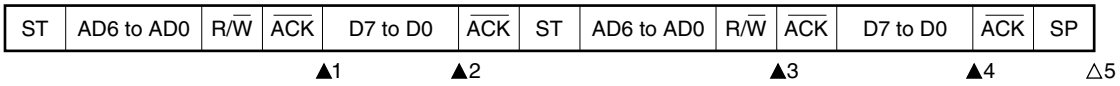


▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 0001×110B
▲4: IICS = 0001××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

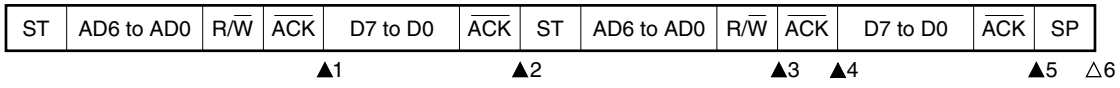
(i) When WTIM = 0 (after restart, does not match address (= extension code))



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0010×010B
▲4: IICS = 0010×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, does not match address (= extension code))

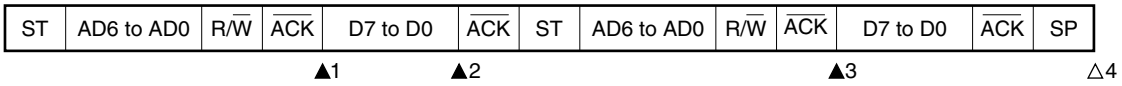


▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 0010×010B
▲4: IICS = 0010×110B
▲5: IICS = 0010××00B
△6: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM = 0 (after restart, does not match address (= not extension code))

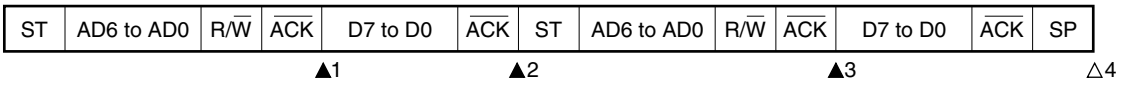


▲1 ▲2 ▲3 △4

▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 00000110B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, does not match address (= not extension code))



▲1 ▲2 ▲3 △4

▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 00000110B
△4: IICS = 00000001B

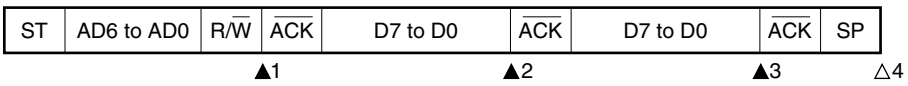
Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(3) Slave device operation (when receiving extension code)

The device is always participating in communication when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

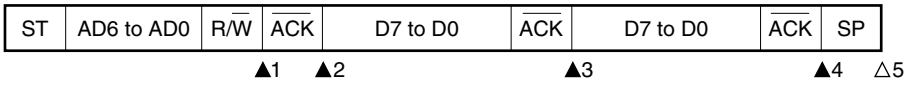
(i) When WTIM = 0



▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0010×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

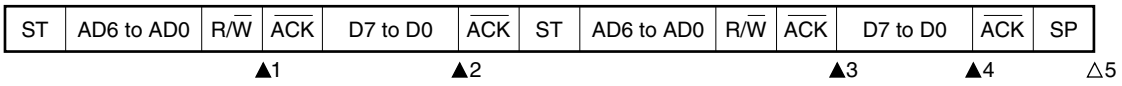


▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010×100B
▲4: IICS = 0010××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

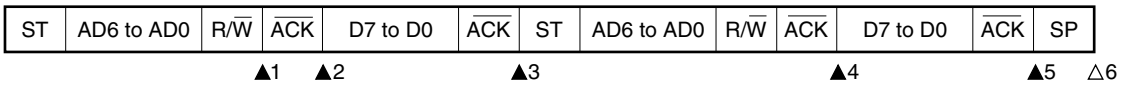
(i) When WTIM = 0 (after restart, matches SVA)



▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0001×110B
▲4: IICS = 0001×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, matches SVA)

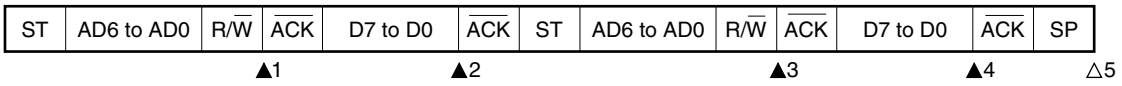


▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010××00B
▲4: IICS = 0001×110B
▲5: IICS = 0001××00B
△6: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

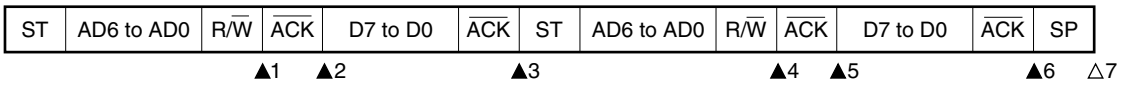
(i) When WTIM = 0 (after restart, extension code reception)



▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0010×010B
▲4: IICS = 0010×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, extension code reception)



▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010××00B
▲4: IICS = 0010×010B
▲5: IICS = 0010×110B
▲6: IICS = 0010××00B
△7: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

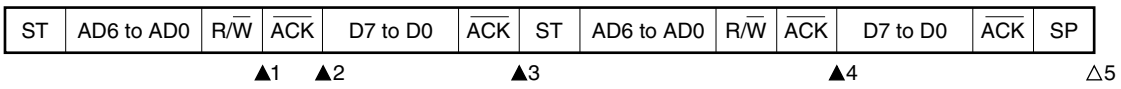
(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM = 0 (after restart, does not match address (= not extension code))

▲1: IICS = 00100010B
▲2: IICS = 00100000B
▲3: IICS = 00000110B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, does not match address (= not extension code))

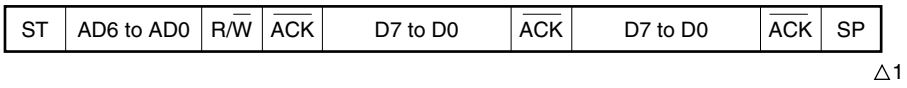


▲1: IICS = 00100010B
▲2: IICS = 00100110B
▲3: IICS = 00100×00B
▲4: IICS = 00000110B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICS = 00000001B

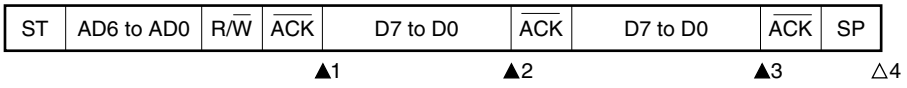
Remark △: Generated only when SPIE = 1

(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 MSTS bit each time interrupt request signal INTIICA has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

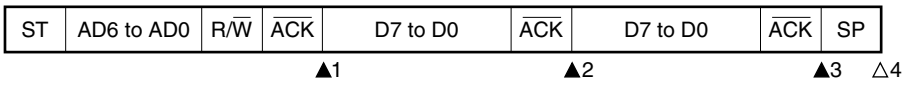
(i) When WTIM = 0



▲1: IICS = 0101×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

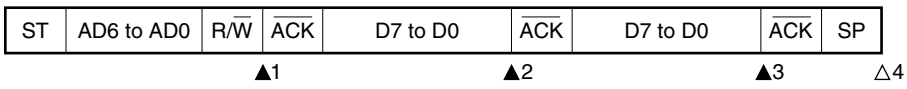


▲1: IICS = 0101×110B
▲2: IICS = 0001×100B
▲3: IICS = 0001××00B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

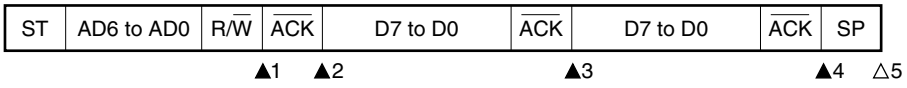
(i) When WTIM = 0



▲1: IICS = 0110×010B
▲2: IICS = 0010×000B
▲3: IICS = 0010×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



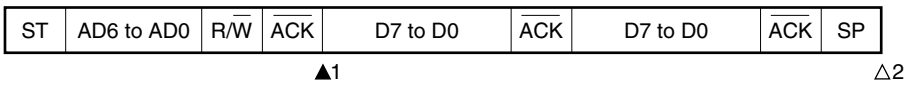
▲1: IICS = 0110×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010×100B
▲4: IICS = 0010××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS bit each time interrupt request signal INTIICA has occurred to check the arbitration result.

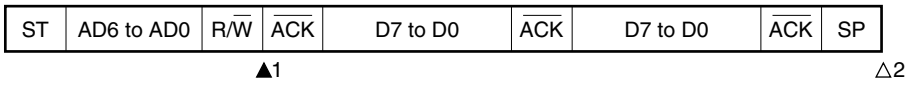
(a) When arbitration loss occurs during transmission of slave address data (when WTIM = 1)



▲1: IICS = 01000110B
△2: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1

(b) When arbitration loss occurs during transmission of extension code



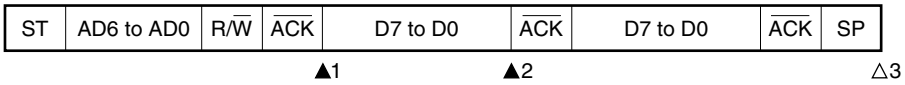
▲1 △2

▲1: IICS = 0110×010B
Sets LREL = 1 by software
△2: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

(i) When WTIM = 0

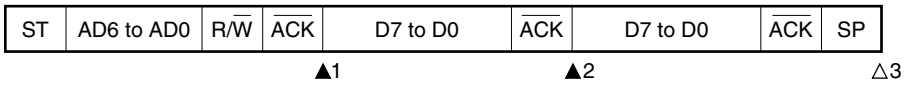


▲1 ▲2 △3

▲1: IICS = 10001110B
▲2: IICS = 01000000B
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1

(ii) When WTIM = 1

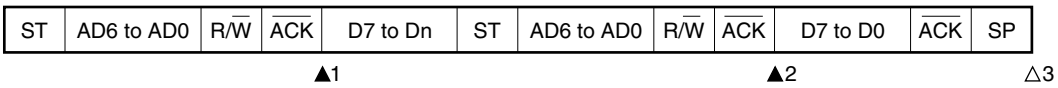


▲1: IICS = 10001110B
▲2: IICS = 01000100B
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1

(d) When loss occurs due to restart condition during data transfer

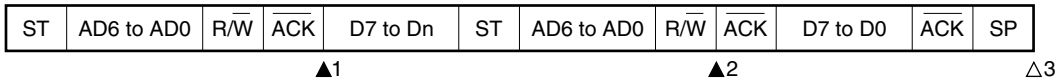
(i) Not extension code (Example: unmatches with SVA)



▲1: IICS = 1000×110B
▲2: IICS = 01000110B
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care
n = 6 to 0

(ii) Extension code



▲1: IICS = 1000×110B
▲2: IICS = 01100010B
Sets LREL = 1 by software
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care
n = 6 to 0

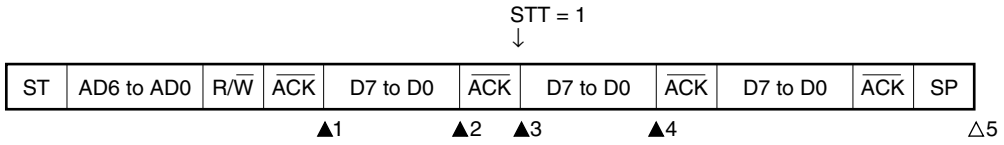
(e) When loss occurs due to stop condition during data transfer

▲1: IICS = 10000110B
△2: IICS = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care
n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

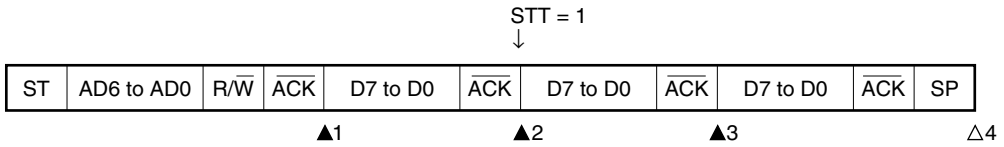
(i) When WTIM = 0



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets the WTIM bit to 1)
▲3: IICS = 1000×100B (Clears the WTIM bit to 0)
▲4: IICS = 01000000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

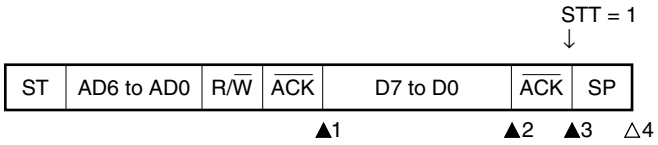


▲1: IICS = 1000×110B
▲2: IICS = 1000×100B (Sets the STT bit to 1)
▲3: IICS = 01000100B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

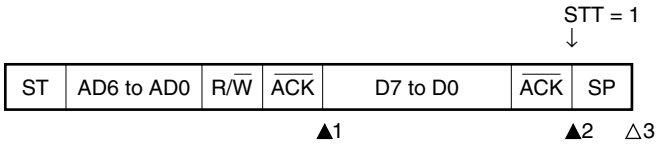
(i) When WTIM = 0



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets the WTIM bit to 1)
▲3: IICS = 1000××00B (Sets the STT bit to 1)
△4: IICS = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

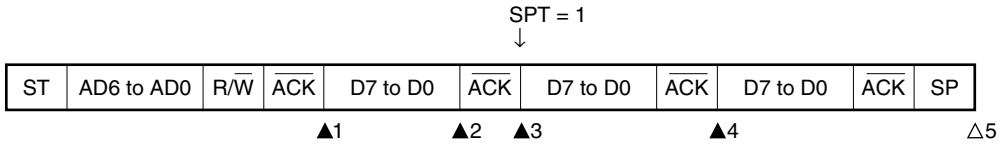


▲1: IICS = 1000×110B
▲2: IICS = 1000××00B (Sets the STT bit to 1)
△3: IICS = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

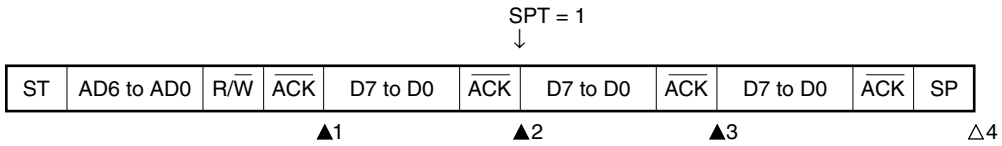
(i) When WTIM = 0



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets the WTIM bit to 1)
▲3: IICS = 1000×100B (Clears the WTIM bit to 0)
▲4: IICS = 01000100B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



▲1: IICS = 1000×110B
▲2: IICS = 1000×100B (Sets the SPT bit to 1)
▲3: IICS = 01000100B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

12.6 Timing Charts

When using the I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC bit (bit 3 of the IICA status register (IICS)), which specifies the data transfer direction, and then starts serial communication with the slave device.

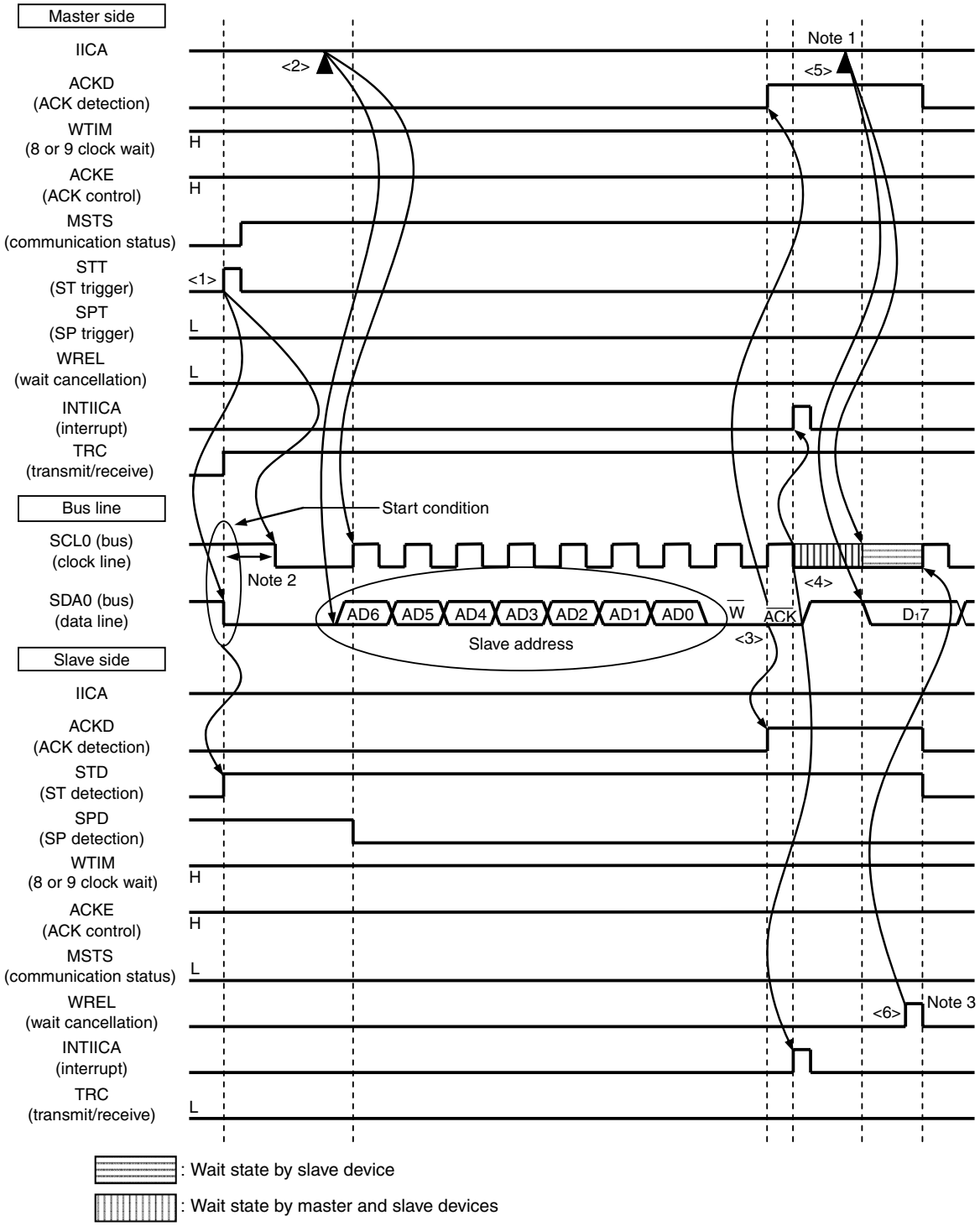
Figures 12-32 and 12-33 show timing charts of the data communication.

The IICA shift register (IICA)'s shift operation is synchronized with the falling edge of the serial clock (SCL0). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA0 pin.

Data input via the SDA0 pin is captured into IICA at the rising edge of SCL0.

Figure 12-32. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/4)

(1) Start condition ~ address ~ data



Notes 1. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during master transmission.

2. Make sure that the time between the fall of the SDA0 pin signal and the fall of the SCL0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 12-32 are explained below.

<1> The start condition trigger is set by the master device (STT = 1) and a start condition (SDA0 = 0 and SCL0 = 1) is generated once the bus data line goes low (SDA0 = 0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS = 1). The master device is ready to communicate once the bus clock line goes low (SCL0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register (IICA) and transmits the slave address.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The master device writes the data to transmit to the IICA register and releases the wait status that it set by the master device.

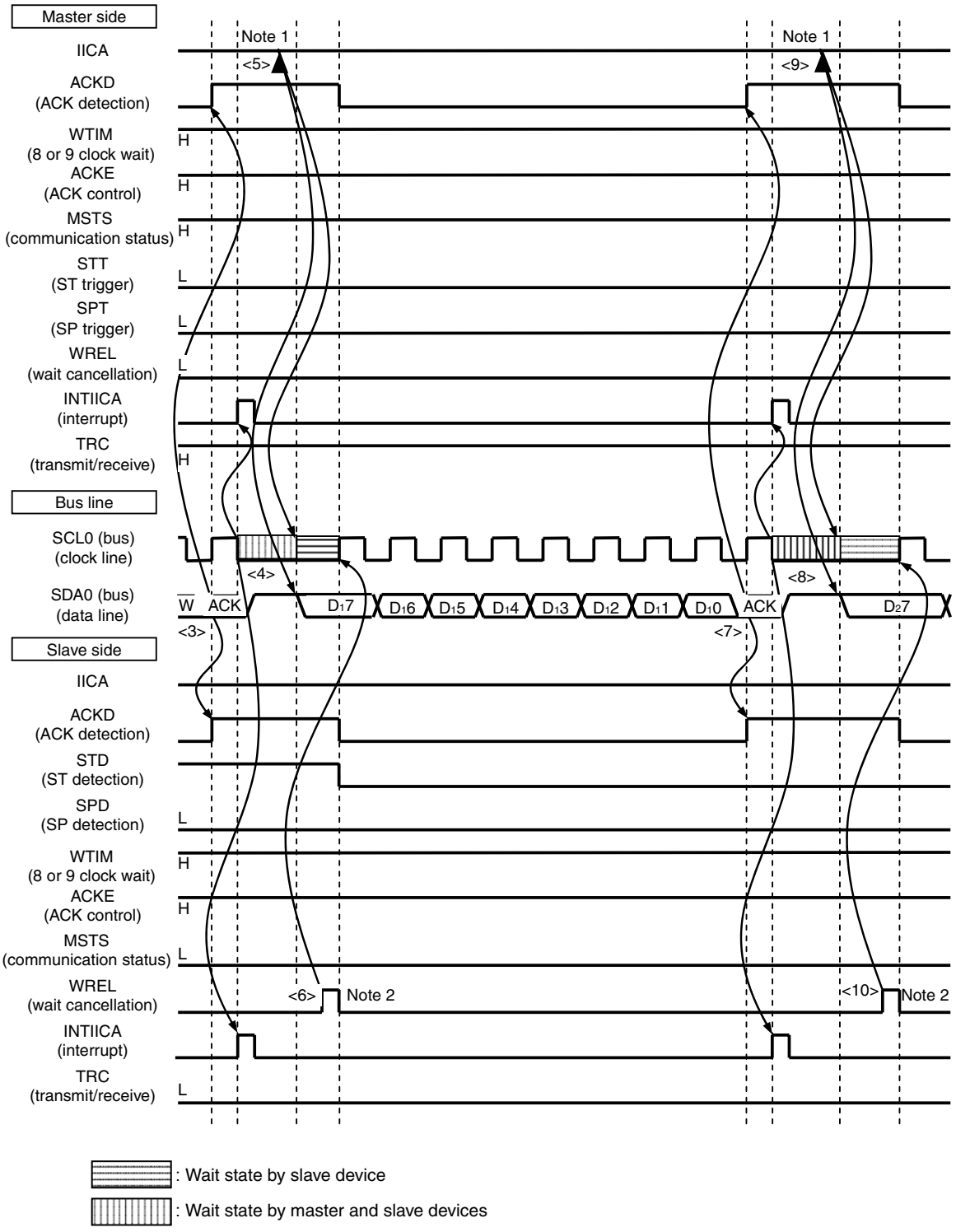
<6> If the slave device releases the wait status (WREL = 1), the master device starts transferring data to the slave device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 12-32 represent the entire procedure for communicating data using the I^2C bus. Figure 12-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 12-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 12-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 12-32. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/4)

(2) Address ~ data ~ data



Notes 1. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 12-32 are explained below.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The master device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the master device.

<6> If the slave device releases the wait status (WREL = 1), the master device starts transferring data to the slave device.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<9> The master device writes the data to transmit to the IICA register and releases the wait status that it set by the master device.

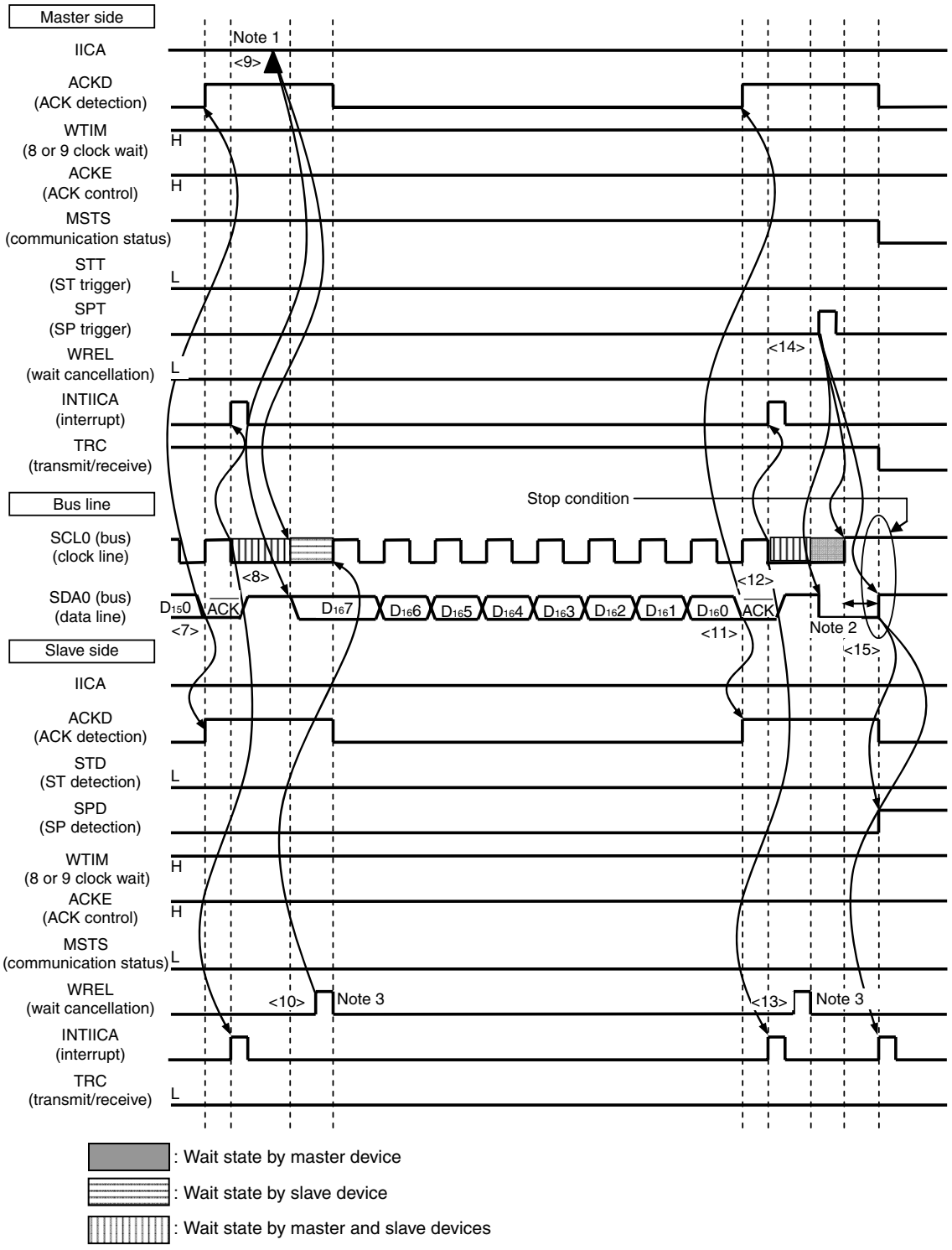
<10> The slave device reads the received data and releases the wait status (WREL = 1). The master device then starts transferring data to the slave device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 12-32 represent the entire procedure for communicating data using the I^2C bus. Figure 12-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 12-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 12-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 12-32. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/4)

(3) Data ~ data ~ Stop condition



Notes 1. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during master transmission.

2. Make sure that the time between the rise of the SCL0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 12-32 are explained below.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the master device.

<10> The slave device reads the received data and releases the wait status (WREL = 1). The master device then starts transferring data to the slave device.

<11> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<12> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<13> The slave device reads the received data and releases the wait status (WREL = 1).

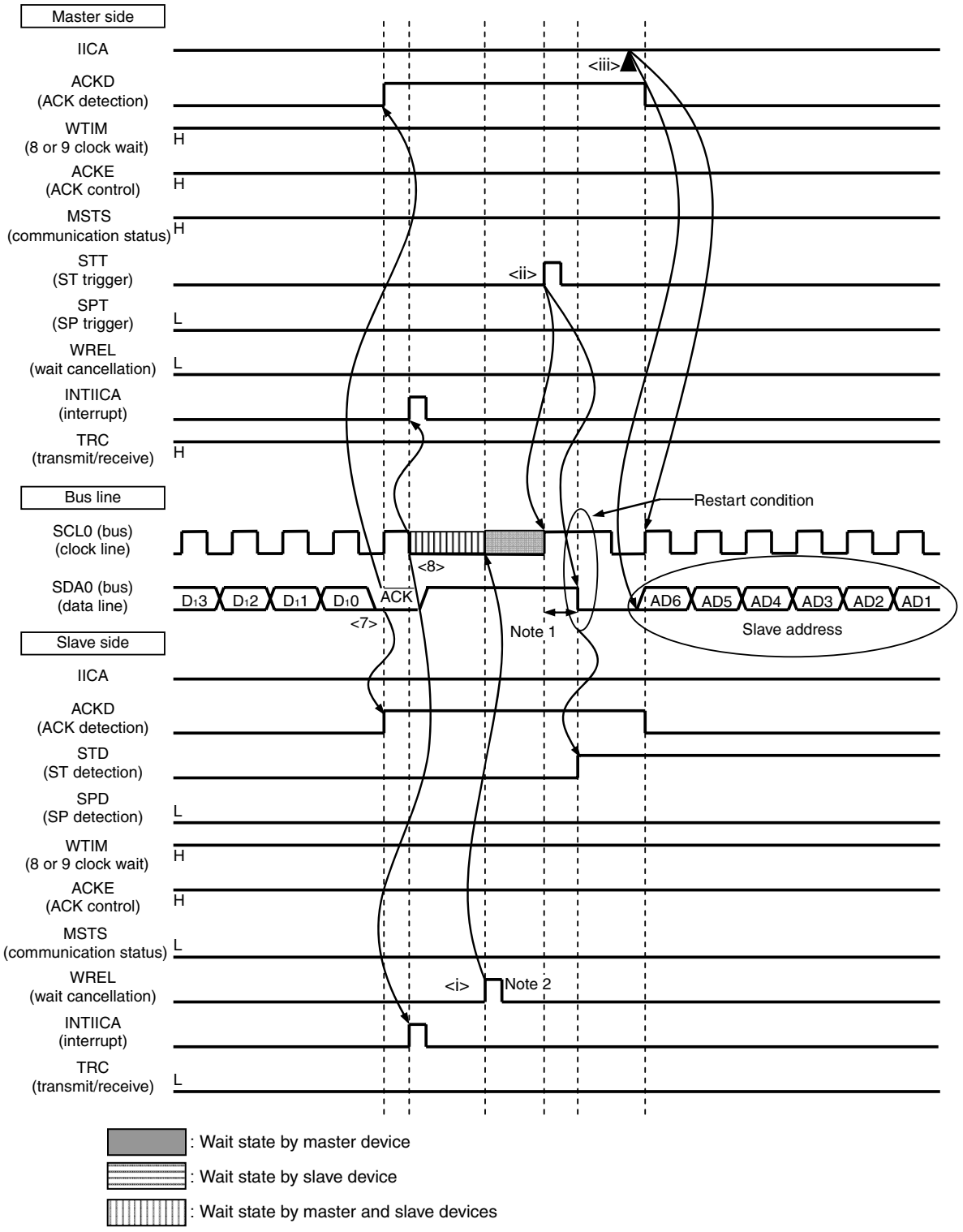
<14> After a stop condition trigger is set, the bus data line is cleared (SDA0 = 0) and the bus clock line is set (SCL0 = 1). The stop condition is then generated by setting the bus data line (SDA0 = 1) after the stop condition setup time has elapsed.

<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICA: stop condition).

Remark <1> to <15> in Figure 12-32 represent the entire procedure for communicating data using the I^2C bus. Figure 12-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 12-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 12-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 12-32. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (4/4)

(4) Data ~ restart condition ~ address



Notes 1. Make sure that the time between the rise of the SCL0 pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

2. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The following describes the operations in Figure 12-32 (4) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <1> to <3> are performed. These steps return the processing to step <3>, the data transmission step.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

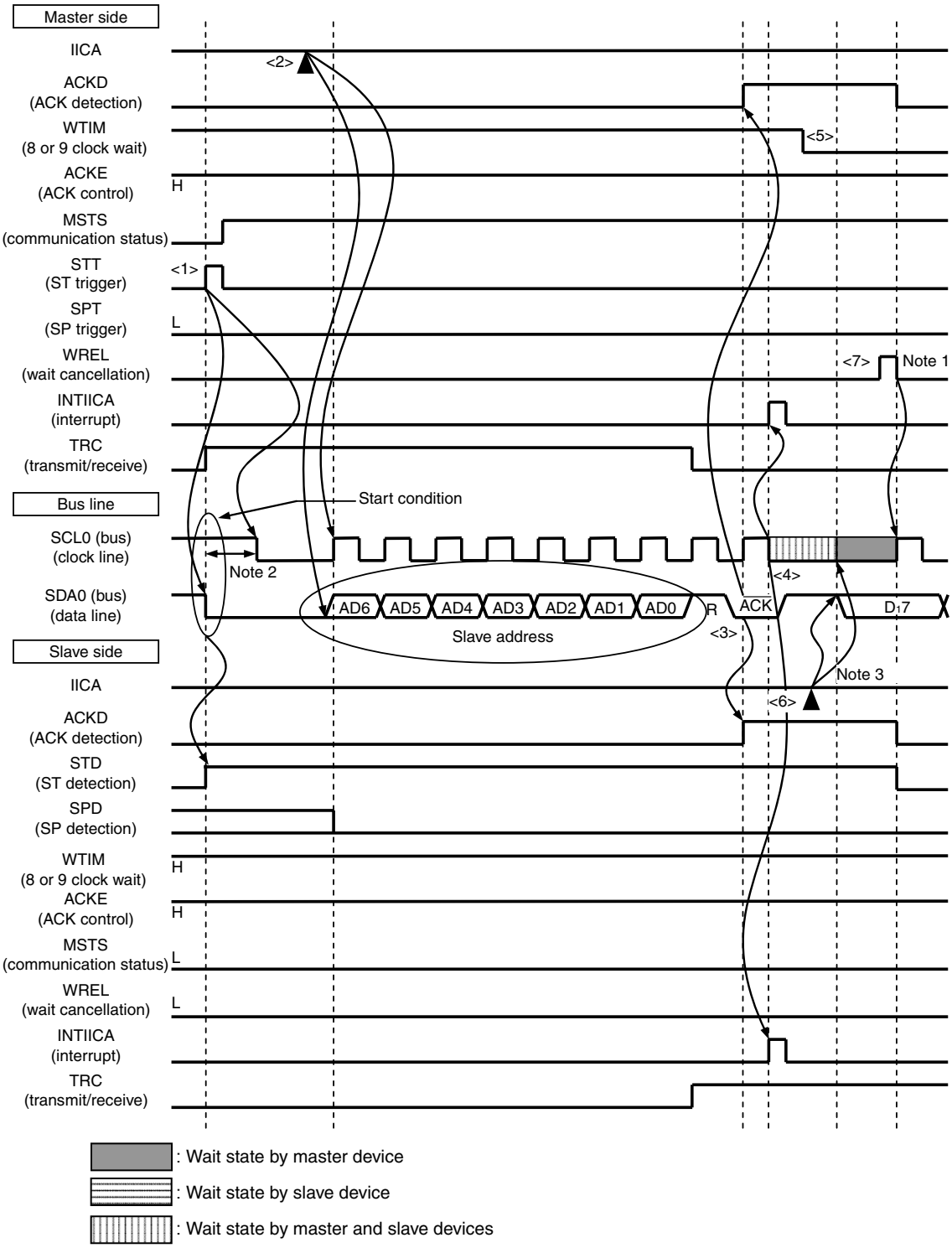
<i> The slave device reads the received data and releases the wait status (WREL = 1).

<ii> The start condition trigger is set again by the master device (STT = 1) and a start condition (SDA0 = 0 and SCL0 = 1) is generated once the bus clock line goes high (SCL0 = 1) and the bus data line goes low (SDA0 = 0) after the restart condition setup time has elapsed. When the start condition is subsequently detected, the master device is ready to communicate once the bus clock line goes low (SCL0 = 0) after the hold time has elapsed.

<iii> The master device writes the address + R/W (transmission) to the IICA shift register (IICA) and transmits the slave address.

Figure 12-33. 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 ~ data



Notes 1. To cancel master wait, write "FFH" to IICA or set the WREL bit.

2. Make sure that the time between the fall of the SDA0 pin signal and the fall of the SCL0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during slave transmission.

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 12-33 are explained below.

<1> The start condition trigger is set by the master device (STT = 1) and a start condition (SDA0 = 0 and SCL0 = 1) is generated once the bus data line goes low (SDA0 = 0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS = 1). The master device is ready to communicate once the bus clock line goes low (SCL0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register (IICA) and transmits the slave address.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIM = 0).

<6> The slave device writes the data to transmit to the IICA register and releases the wait status that it set by the slave device.

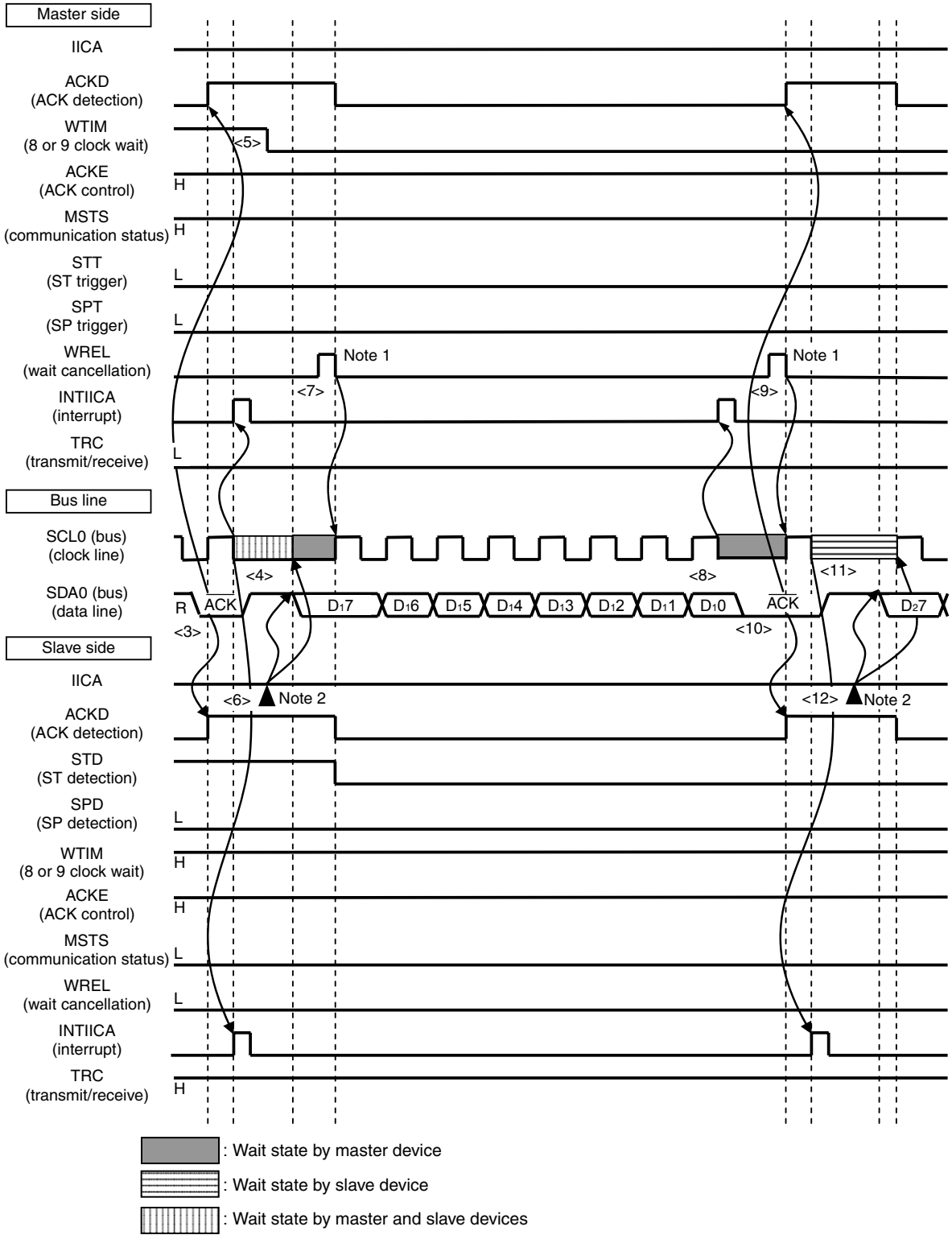
<7> If the master device releases the wait status (WREL = 1), the slave device starts transferring data to the master device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 12-33 represent the entire procedure for communicating data using the I^2C bus. Figure 12-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 12-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 12-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 12-33. Example of Slave to Master Communication (When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)

(2) Address ~ data ~ data



Notes 1. To cancel master wait, write “FFH” to IICA or set the WREL bit.

2. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during slave transmission.

The meanings of <3> to <12> in (2) Address ~ data ~ data in Figure 12-33 are explained below.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIM = 0).

<6> The slave device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the slave device.

<7> If the master device releases the wait status (WREL = 1), the slave device starts transferring data to the master device.

<8> The master device sets a wait status (SCL0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA: end of transfer). The master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the wait status (WREL = 1).

<10> The ACK is detected by the slave device (ACKD = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA: end of transfer).

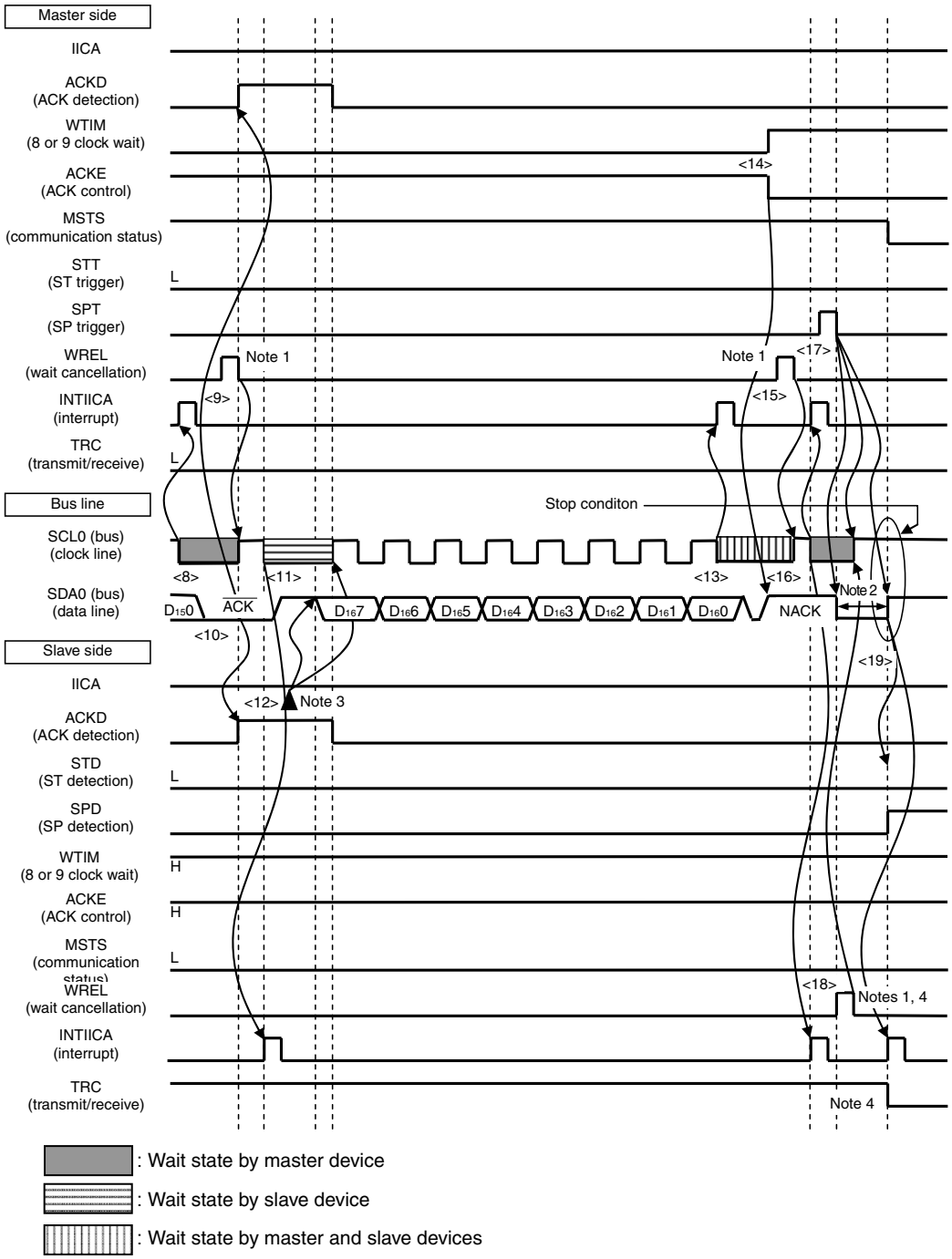
<12> The slave device writes the data to transmit to the IICA register and releases the wait status that it set by the slave device. The slave device then starts transferring data to the master device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 12-33 represent the entire procedure for communicating data using the I^2C bus. Figure 12-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 12-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 12-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 12-33. 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) Data ~ data ~ stop condition



Notes 1. To cancel a wait state, write “FFH” to IICA or set the WREL bit.

2. Make sure that the time between the rise of the SCL0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during slave transmission.

4. If a wait state during slave transmission is canceled by setting the WREL bit, the TRC bit will be cleared.

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 12-33 are explained below.

<8> The master device sets a wait status (SCL0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA: end of transfer). The master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the wait status (WREL = 1).

<10> The ACK is detected by the slave device (ACKD = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA: end of transfer).

<12> The slave device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the slave device. The slave device then starts transferring data to the master device.

<13> The master device issues an interrupt (INTIICA: end of transfer) at the falling edge of the 8th clock, and sets a wait status (SCL0 = 0). Because ACK control (ACKE = 1) is performed, the bus data line is at the low level (SDA0 = 0) at this stage.

<14> The master device sets NACK as the response (ACKE = 0) and changes the timing at which it sets the wait status to the 9th clock.

<15> If the master device releases the wait status (WREL = 1), the slave device detects the NACK (ACK = 0) at the rising edge of the 9th clock.

<16> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<17> When the master device issues a stop condition (SPT = 1), the bus data line is cleared (SDA0 = 0) and the master device releases the wait status. The master device then waits until the bus clock line is set (SCL0 = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the wait status (WREL = 1) to end communication. Once the slave device releases the wait status, the bus clock line is set (SCL0 = 1).

<19> Once the master device recognizes that the bus clock line is set (SCL0 = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDA0 = 1) and issues a stop condition. The slave device detects the generated stop condition and both the master device and slave device issue an interrupt (INTIICA: stop condition).

Remark <1> to <19> in Figure 12-33 represent the entire procedure for communicating data using the I^2C bus. Figure 12-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 12-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 12-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

CHAPTER 13 CEC TRANSMISSION/RECEPTION CIRCUIT

13.1 Functions of CEC Transmission/Reception Circuit

The CEC transmission/reception circuit can generate and receive CEC signals conforming to the CEC (Consumer Electronics Control) standard, as well as automatically detect communication statuses by hardware. CEC transmission/reception can be easily controlled by using these functions.

- Serial communication conforming to the CEC standard can be performed.
- The operating clock can be selected from the main system clock and subsystem clock.
- The low-level width/bit width of the start bit and data bit can be set to different values for transmission and reception.
- Errors and communication statuses can be detected by hardware.
- Signal-free time can be counted.

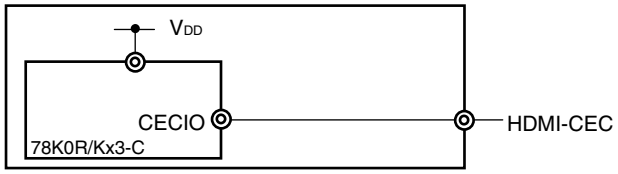
<R> **(1) About the setting of each mode**

(a) TYPE1: When using the internal pull-up resistors and diodes

- Using P62
- Using the power supply at V_{DD} = 3.3 V±10%
- Not needing external circuits to the microcontroller.

P62	PM62	PF62	PIM62	PU62
1	0	1	1	1

PF62 = 1: CECIO mode
PIM62 = 1: CEC input buffer
PU62 = 1: Connects pull-up resistor + diode

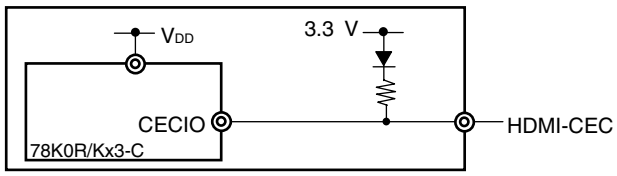


(b) TYPE2: When using the circuit connecting to external pull-up resistors and diodes

- Using P62
- Using the power supply at V_{DD} = 3.3 V±10%
- Needing to connect external pull-up resistors and diodes to the microcontroller

P62	PM62	PF62	PIM62	PU62
1	0	1	1	0

PF62 = 1: CECIO mode
PIM62 = 1: CEC input buffer
PU62 = 0: Do not connected pull-up resistor + diode

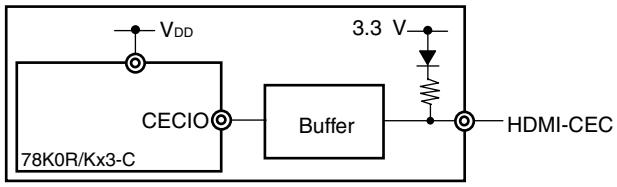


(c) TYPE3: When using the CEC function out of range of V_{DD} = 3.3 V±10%

- Using P62
- Requiring the external circuit to convert the CECIO signal voltage to that on CEC standard value
- Requiring the external circuit to satisfy the electrical characteristics of normal input buffer, before the signal being inputted to the CECIO pin
- This type can be used at V_{DD} = 2.7 to 5.5 V

P62	PM62	PF62	PIM62	PU62
1	0	1	0	0

PF62 = 1: CECIO mode
PIM62 = 0: Normal input buffer
PU62 = 0: Dose not connected pull-up resistor + diode



(d) TYPE4: When using the CEC function with connecting to CEC input and output separately

- Using P110 and P111
- Being used when the CEC functions are used separately divided into CEC input and output
- Requiring the external circuit to satisfy the electrical characteristics of normal input buffer, before the signal being inputted to the CECIN pin
- Requiring the external circuit to convert the CECOUT signal voltage to that on CEC standard value
- This type can be used at V_{DD} = 2.7 to 5.5 V

P110	PM110	P111	PM111	PF110	PF111
–	1	1	0	1	1

PF110 = 1: CECIN mode
PF111 = 1: CECOUT mode

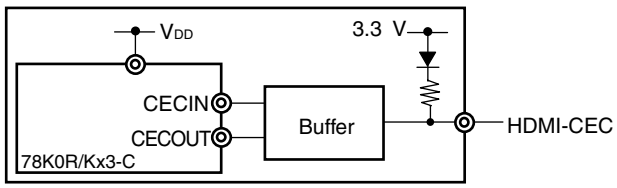
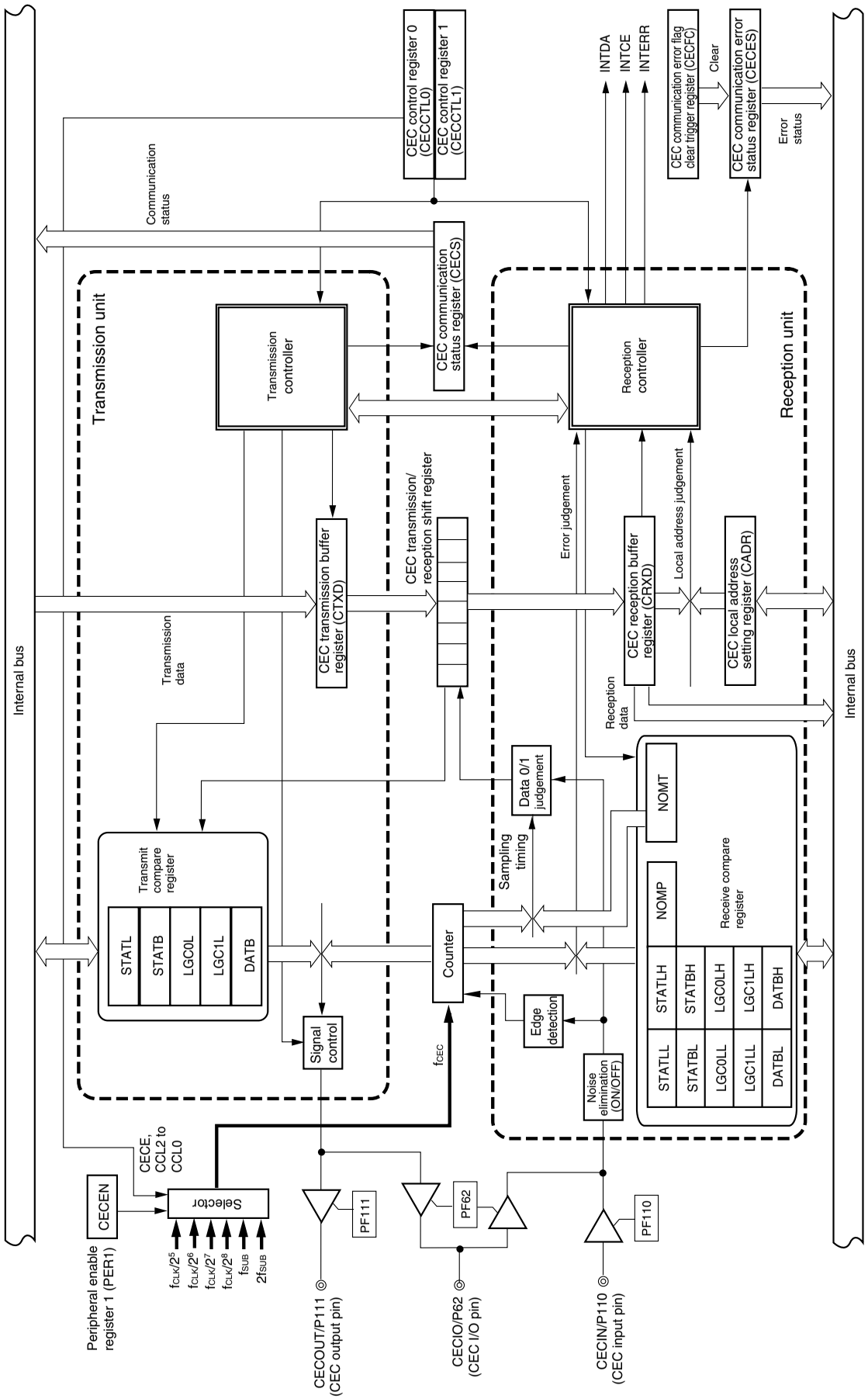


Figure 13-1. Block Diagram of CEC transmission/reception circuit



13.2 Configuration of CEC Transmission/Reception Circuit

The CEC transmission/reception circuit includes the following hardware.

Table 13-1. Configuration of CEC Transmission/Reception Circuit

Item	Configuration
Control registers	Peripheral enable register 1 (PER1) CEC control register 0 (CECCTL0) CEC control register 1 (CECCTL1) CEC communication status register (CECS) CEC communication error status register (CECES) CEC communication error flag clear trigger register (CECFC) CEC local address setting register (CADR) Port function register 6 (PF6) Port function register 11 (PF11) Port register 6 (P6) Port mode register 6 (PM6)
Compare registers[Notes 1, 2, 3]	CEC reception buffer register (CRXD) CEC transmission buffer register (CTXD) CEC reception start bit minimum low width setting register (STATLL) CEC reception start bit maximum low width setting register (STATLH) CEC reception start bit minimum bit width setting register (STATBL) CEC reception start bit maximum bit width setting register (STATBH) CEC reception logical 0 minimum low width setting register (LGC0LL) CEC reception logical 0 maximum low width setting register (LGC0LH) CEC reception logical 1 minimum low width setting register (LGC1LL) CEC reception logical 1 maximum low width setting register (LGC1LH) CEC reception data bit minimum bit width setting register (DATBL) CEC reception data bit maximum bit width setting register (DATBH) CEC transmission start bit low width setting register (STATL) CEC transmission start bit width setting register (STATB) CEC transmission logical 0 low width setting register (LGC0L) CEC transmission logical 1 low width setting register (LGC1L) CEC transmission data bit width setting register (DATB) CEC data bit reference width setting register (NOMP) CEC reception data sampling time setting register (NOMT)

Notes 1. 0 can be written to the registers, but the CEC transmission/reception circuit will not operate normally.

2. Set up all registers other than buffer registers even when only receiving or transmitting data.

3. Some of the register setting values must have a specific relationship in size. Set the registers such that the following relationships are observed.

- STATL < STATB
- LGC1L < LGC0L < DATB
- STATLL < STATLH
- STATBL < STATBH
- LGC0LL < LGC0LH
- LGC1LL < LGC1LH
- DATBL < DATBH

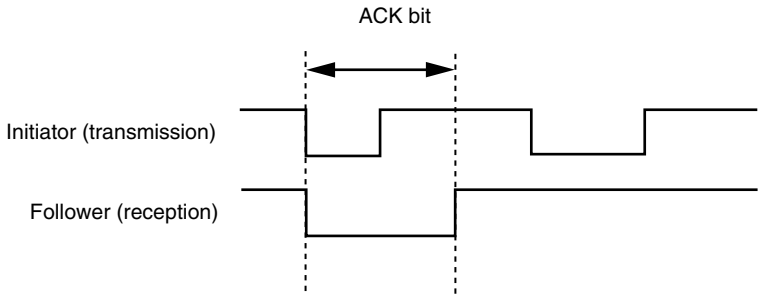
13.3 Term Description

- Initiator: Device that transmits or is transmitting CEC messages
- Follower: Device that receives or is receiving CEC messages
- Message: All data from the start bit to the operand
- Initiator address: Source address
- Destination address: Destination address
- Direct address communication (direct address message): Communication with one follower
- Broadcast communication (direct address message): Communication with multiple followers
- Arbitration: Prioritizing the devices that output a low level to the CEC line when multiple initiators exist
- Arbitration loss: State in which competing devices are prioritized. At this time, the local station stops transmitting.
- Bus free: State in which no communication is performed and transmission can be performed
- Bus busy: During communication
- Error handling: Outputting an error pulse (low level with a width of the bit width × 1.5) and transitioning to the communication standby state when a bit width shorter than the bit width of the data bit is received

The logic levels received at the ACK bit timing are as follows.

- ACK: Outputs logical 0.
- NACK: Outputs logical 1.

Example: If the initiator outputs logical 1 and the follower outputs logical 0 during an ACK bit period, the initiator transmits a NACK and the follower transmits an ACK.



13.4 Registers Controlling Transmission/Reception Circuit

The CEC transmission/reception circuit is controlled by the following registers.

- Peripheral enable register 1 (PER1)
- CEC local address setting register (CADR)
- CEC transmission buffer register (CTXD)
- CEC reception buffer register (CRXD)
- CEC control register 0 (CECCTL0)
- CEC control register 1 (CECCTL1)
- CEC communication status register (CECS)
- CEC communication error status register (CECES)
- CEC communication error flag clear trigger register (CECFC)
- Port function register 6 (PF6)
- Port function register 11 (PF11)

(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.

Be sure to set bit 4 (CECEN) of this register to 1 when using the CEC transmission/reception circuit.

PER1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13-2. Format of Peripheral Enable Register 1 (PER1)

Address: F00F1H After reset: 00H R/W

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

CECEN	CEC transmission/reception circuit input clock control
0	Stops supply of input clock. • SFRs used by CEC transmission/reception circuit cannot be written. • CEC transmission/reception circuit is in reset state.
1	Supplies input clock. • SFRs used by CEC transmission/reception circuit can be read/written.

Cautions 1. When setting the CEC transmission/reception circuit, be sure to set CECEN to 1 first. If CECEN = 0, writing to a control register of the CEC transmission/reception circuit is ignored, and, even if the register is read, only the default value is read.

2. Be sure to clear bits 0 to 3, 6, and 7 of the PER1 register to 0.

(2) CEC local address setting register (CADR)

CADR is a 16-bit register that sets local addresses. It is valid only during a reception, the ADR00 to ADR14 bits correspond to CEC logical addresses 0 to 14, and up to 15 local addresses can be set. To specify address 15 as the CEC local address, clear the ADR00 to ADR14 bits to 0. A broadcast address always operates as a local address.

For example, when using addresses 0 as local addresses, 1 is set to the ADR00 bits.

CADR can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 13-3. Format of CEC Local Address Setting Register (CADR) (1/2)

Address: F0300H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CADR	0	ADR 14	ADR 13	ADR 12	ADR 11	ADR 10	ADR 09	ADR 08	ADR 07	ADR 06	ADR 05	ADR 04	ADR 03	ADR 02	ADR 01	ADR 00

ADR00	Address 0 (TV)
0	Does not set as local address.
1	Sets as local address.

ADR01	Address 1 (recording device 1)
0	Does not set as local address.
1	Sets as local address.

ADR02	Address 2 (recording device 2)
0	Does not set as local address.
1	Sets as local address.

ADR03	Address 3 (tuner 1)
0	Does not set as local address.
1	Sets as local address.

ADR04	Address 4 (playback device 1)
0	Does not set as local address.
1	Sets as local address.

ADR05	Address 5 (audio system)
0	Does not set as local address.
1	Sets as local address.

ADR06	Address 6 (tuner 2)
0	Does not set as local address.
1	Sets as local address.

Figure 13-3. Format of CEC Local Address Setting Register (CADR) (2/2)

Address: FF2AH, FF2BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CADR	0	ADR 14	ADR 13	ADR 12	ADR 11	ADR 10	ADR 09	ADR 08	ADR 07	ADR 06	ADR 05	ADR 04	ADR 03	ADR 02	ADR 01	ADR 00

ADR07	Address 7 (tuner 3)
0	Does not set as local address.
1	Sets as local address.

ADR08	Address 8 (playback device 2)
0	Does not set as local address.
1	Sets as local address.

ADR09	Address 9 (recording device 3)
0	Does not set as local address.
1	Sets as local address.

ADR10	Address 10 (tuner 4)
0	Does not set as local address.
1	Sets as local address.

ADR11	Address 11 (playback device 3)
0	Does not set as local address.
1	Sets as local address.

ADR12	Address 12 (reserved)
0	Does not set as local address.
1	Sets as local address.

ADR13	Address 13 (reserved)
0	Does not set as local address.
1	Sets as local address.

ADR14	Address 14 (specific use)
0	Does not set as local address.
1	Sets as local address.

Caution To specify address 15 (unregistered) as the CEC local address, clear the ADR00 to ADR14 bits to 0.

(3) CEC transmission buffer register (CTXD)

CTXD is an 8-bit register that sets transmit data. It sequentially transmits eight bits of data, starting from bit 7. A transmission/reception interrupt request signal (INTDA) is generated at the start timing of the header block and data block. Successive transmission can be performed by writing the next data to CTXD before the transmission ends after INTDA was generated.

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

Reset signal generation clears this register to 00H.

Figure 13-4. Format of CEC Transmission Buffer Register (CTXD)

Address: FFF78H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTXD	CTXD							

Caution If an underrun error occurs (UERR = 1), transmission is not continued. An error interrupt is generated and the transmission wait state is entered.

(4) CEC reception buffer register (CRXD)

CRXD is an 8-bit register that retains receive data.

Receive data can be read by reading this register.

For every byte of data received, new data will be transferred from the CEC reception shift register.

CRXD can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13-5. Format of CEC Reception Buffer Register (CRXD)

Address: FFF79H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
CRXD	CRXD							

Caution If an overrun error occurs (OERR = 1), the data of the reception buffer register will be overwritten.

(5) CEC control register 0 (CECCTL0)

CECCTL0 selects enabling operation, starting transmission, and the operating clock.

CECCTL0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13-6. Format of CEC Control Register 0 (CECCTL0) (1/2)

Address: FFF7DH After reset: 00H R/W

Symbol	<7>	<6>	5	4	3	<2>	<1>	<0>
CECCTL0	CECE	ACKTEN	CCL2	CCL1	CCL0	TXTRG	CECRXEN	EOM

CECE	CEC operation enable flag
0	Stops CEC operation. * Does not reset control register even if CECE = 0 is set.
1	Enables CEC transmit/receive operation.

ACKTEN	Enabling ACK bit timing error (bit width) check
0	Does not detect ACK bit timing errors (bit width).
1	Detects ACK bit timing errors (bit width). Notes 1

CCL2	CCL1	CCL0	Source clock (f_{CEC}) selection Notes 2, 3
0	0	0	$f_{CLK}/2^5$
0	0	1	$f_{CLK}/2^6$
0	1	0	$f_{CLK}/2^7$
0	1	1	$f_{CLK}/2^8$
0	0	0	f_{SUB} (32.768 kHz)
0	0	1	$f_{SUB} \times 2$ (32.768 kHz × 2)
Other than above			Setting prohibited

Notes 1. Timing errors are detected for the bit width (the values specified for DATBL and DATBH), in addition to the low-level width of the ACK bit (the values specified for LGC0LL, LGC0LH, LGC1LL, and LGC1LH). However, the maximum bit width (DATBH) is not checked for the ACK bit of the last frame (EOM = 1), even if ACKTEN is 1.

2. Rewritable only when CECE = 0.

3. Set the source clock frequency of the CEC counter to a frequency from 7.8125 to 78.125 kHz. Examples of CEC counter source clock settings are shown below.

CEC Operating Clock	CEC Counter Source Clock (f_{CEC})					
	When f_{CLK} = 20 MHz	When f_{CLK} = 16 MHz	When f_{CLK} = 12 MHz	When f_{CLK} = 10 MHz	When f_{CLK} = 8 MHz	When f_{CLK} = 2 MHz
$f_{CLK}/2^5$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	62.5 kHz
$f_{CLK}/2^6$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	31.25 kHz
$f_{CLK}/2^7$	Setting prohibited	Setting prohibited	Setting prohibited	78.125 kHz	62.5 kHz	15.625 kHz
$f_{CLK}/2^8$	78.125 kHz	62.5 kHz	46.875 kHz	39.0625 kHz	31.25 kHz	7.8125 kHz
f_{SUB}	32.768 kHz (when operating at f_{SUB} = 32.768 kHz)					
$f_{SUB} \times 2$	65.536 kHz (when operating at f_{SUB} = 32.768 kHz)					

Remarks 1. f_{CEC}: CEC counter source clock (selected by bits 5 to 3 (CCL2 to CCL0))

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

3. f_{SUB}: Subsystem clock oscillation frequency

Figure 13-6. Format of CEC Control Register 0 (CECCTL0) (2/2)

Address: FFF7DH After reset: 00H R/W

Symbol	<7>	<6>	5	4	3	<2>	<1>	<0>
CECCTL0	CECE	ACKTEN	CCL2	CCL1	CCL0	TXTRG	CECRXEN	EOM

TXTRG	Transmission start trigger bit[Notes 1, 2]
0	Does not start CEC transmission. (0 is always read. Writing 0 has no meaning.)
1	Starts CEC transmission when CECE = 1.

CECRXEN	Reception rejection control bit[Notes 3, 4, 5]		
1	Enables continuing reception or reports normal reception (normally selected).		
	Reception status		ACK/NACK timing output
	During direct address reception (to local station)	Normal reception	ACK
		Timing error occurrence	NACK
	During broadcast address reception	Normal reception	NACK
		Timing error occurrence	ACK
	During direct address reception (to another station)	Not participating in communication (high impedance)	
0	Stops continuing reception or reports abnormal reception.		
	Reception status		ACK/NACK timing output
	During direct address reception (to local station)	Normal reception	NACK
		Timing error occurrence	NACK
	During broadcast address reception	Normal reception	ACK
		Timing error occurrence	ACK
	During direct address reception (to another station)	Not participating in communication (high impedance)	

EOM	EOM setting bit
0	Continues transmission.
1	Last frame

Notes 1. TXTRG is a trigger bit from which 0 is always read. If the subsystem clock multiplication clock is selected (CCL3 to CCL0 = 1001B), the clock starts operating a maximum of two clocks later. A transmission starts when a maximum of two clocks elapse after 1 is set to this bit.

2. Writing to the TXTRG bit is prohibited until the CEC transmission ends. Transmission completion can be confirmed by checking that the shift register after the trigger bit is set is 0.
Set TXTRG to 1 when the bus is free (BUSST = 0). Transmission starts no more than two f_{CEC} clock cycles after TXTRG is set to 1.

3. Rewriting the CECRXEN bit during a communication (BUSST = 1) is prohibited.

4. If CECRXEN = 0 is set, reception rejection is reported at the next ACK/NACK timing and the communication standby state is entered.

5. Set CECRXEN = 1 after determining the local address (setting the CADR register). Set CECRXEN = 0, therefore, after checking that address match flag ADRF of the CEC communication status register (CECS) is 1.

(6) CEC control register 1 (CECCTL1)

CECCTL1 selects a digital filter, data interrupt generation, a start bit error interrupt, and whether to generate a communication complete interrupt and when to generate it.

CECCTL1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13-7. Format of CEC Control Register 1 (CECCTL1) (1/2)

Address: F0302H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CECCTL1	CDFC	CINTMK	BLERRD	STERRD	CESEL1	CESEL0	SFT1	SFT0

CDFC	Digital filter select bit[Notes 1, 2]
0	Does not use a digital filter.
1	Uses a digital filter and eliminates noise of one cycle of f_{CEC}.

CINTMK	CEC data interrupt (INTDA) generation select register[Notes 1, 3]	
0	Destination address matches local address.	Generates data interrupt (INTDA).
	Destination address does not match local address.	Does not generate data interrupt (INTDA).
1	Destination address matches local address.	Generates data interrupt (INTDA).
	Destination address does not match local address.	

Notes 1. Rewritable only when CECE = 0.

2. Examples of the noise elimination width settings when the digital filter is used are shown below.

CEC Operating Clock (f_{CEC})	Noise Elimination Width by Digital Filter					
	When f_{CLK} = 20 MHz	When f_{CLK} = 16 MHz	When f_{CLK} = 12 MHz	When f_{CLK} = 10 MHz	When f_{CLK} = 8 MHz	When f_{CLK} = 2 MHz
$f_{CLK}/2^5$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	16 μs
$f_{CLK}/2^6$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	32 μs
$f_{CLK}/2^7$	Setting prohibited	Setting prohibited	Setting prohibited	12.8 μs	16 μs	64 μs
$f_{CLK}/2^8$	12.8 μs	16 μs	21.3 μs	25.6 μs	32 μs	128 μs
f_{SUB}	30.5 μs (when operating at f_{SUB} = 32.768 kHz)					
$f_{SUB} \times 2$	15.2 μs (when operating at f_{SUB} = 32.768 kHz)					

3. Whether to generate a data interrupt (INTDA) of the header block when the destination address and local address do not match during a reception can be selected by setting the CINTMK bit. See **13.7.5 CEC reception** for details.

Remarks 1. f_{CEC}: CEC counter source clock (selected by bits 5 to 3 (CCL2 to CCL0) of the CECCTL0 register)

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

3. f_{SUB}: Subsystem clock oscillation frequency

Figure 13-7. Format of CEC Control Register 1 (CECCTL1) (2/2)

Address: F0302H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CECCTL1	CDFC	CINTMK	BLERRD	STERRD	CESEL1	CESEL0	SFT1	SFT0

BLERRD	Bus lock detection select bit[Notes 1, 2]
0	Does not detect sticking of receive data to high or low level.
1	Detects sticking of receive data to high or low level.

STERRD	Start bit error detection select bit[Notes 1, 3]
0	Does not detect timing errors during start bit reception.
1	Detects timing errors during start bit reception (recommended).

CESEL1	CESEL0	Communication complete interrupt (INTCE) generation timing[Note 1]
0	0	Detects EOM = 1 and generates communication complete interrupt once after ACK transmission/reception completion and another time after signal-free time set by SFT1 and SFT0 is detected.
0	1	Detects EOM = 1 and generates communication complete interrupt after ACK transmission/reception completion.
1	0	Generates communication complete interrupt after signal-free time set by SFT1 and SFT0 is detected.
1	1	Setting prohibited

SFT1	SFT0	Data bit width of signal-free time[Note 4]
0	0	Signal-free time of 3-data bit width
0	1	Signal-free time of 5-data bit width
1	0	Signal-free time of 7-data bit width
1	1	Does not detect signal-free time.

Notes 1. Rewritable only when CECE = 0.

2. The bus lock status of the CEC line can be detected by setting BLERRD to 1. If the next falling edge is not input for a period 2.5 times the 1-data bit width set with the NOMP register in a state in which the falling edge of the CEC line is awaited (excluding the communication standby state), an error interrupt (INTERR) is generated and a bus lock error flag (BLERR) is set. Afterward, the communication standby state is entered.

3. This sets whether to detect timing errors during start bit reception.
Timing errors of the start bit can be detected according to the set value of the STATLL, STATLH, STATBL, or STATBH register by setting STERRD = 1. If a timing error occurred, the start bit for which the error occurred is determined to be disabled and the communication standby state is entered. If STERRD is 0, no timing error is detected. All pulses are determined to be start bits.

4. Rewritable only when SFTST of the CECS register is 0.
The bit width for the signal-free time is specified using the NOMP register.

(7) CEC communication status register (CECS)

CECS indicates the CEC communication status.

CECS can be read by an 8-bit memory manipulation instruction.

Clearing CECE to 0 or generating a reset signal clears this register to 00H.

Figure 13-8. Format of CEC Communication Status Register (CECS) (1/2)

Address: FFF7BH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
CECS	SFTST	0	0	ITCEF	EOMF	TXST	BUSST	ADRF

SFTST	Signal-free time rewrite disable report flag
0	Enables rewriting SFT1 and SFT0. SFTST is cleared to 0 in the following cases. • When CECE = 0 • When the SFT1 and SFT0 rewrite disable period (three operating clocks f_{CEC}, maximum) has elapsed
1	Disables rewriting SFT1 and SFT0. SFTST is set to 1 in the following case. • When a write access to the SFT1 and SFT0 bits (CECCTL1 register) is performed

ITCEF	INTCE generation source flag
0	Generates INTCE if the signal-free time has been counted.
1	Generates INTCE if communication ends or an error is detected.
This setting is enabled only if CESEL1 and CESEL0 are cleared to 0. By checking ITCEF after INTCE is generated, it can be determined which source was the generation source. (For details, see **Figure 13-9**.)	

EOMF	EOM flag
0	The EOM bit received immediately before is logically 0.
1	The EOM bit received immediately before is logically 1.

TXST	Transmission status flag
0	During the communication standby state or reception (A follower is running.)
1	During transmission (The initiator is running.)

Figure 13-8. Format of CEC Communication Status Register (CECS) (2/2)

Address: FFF7BH After reset: 00H R

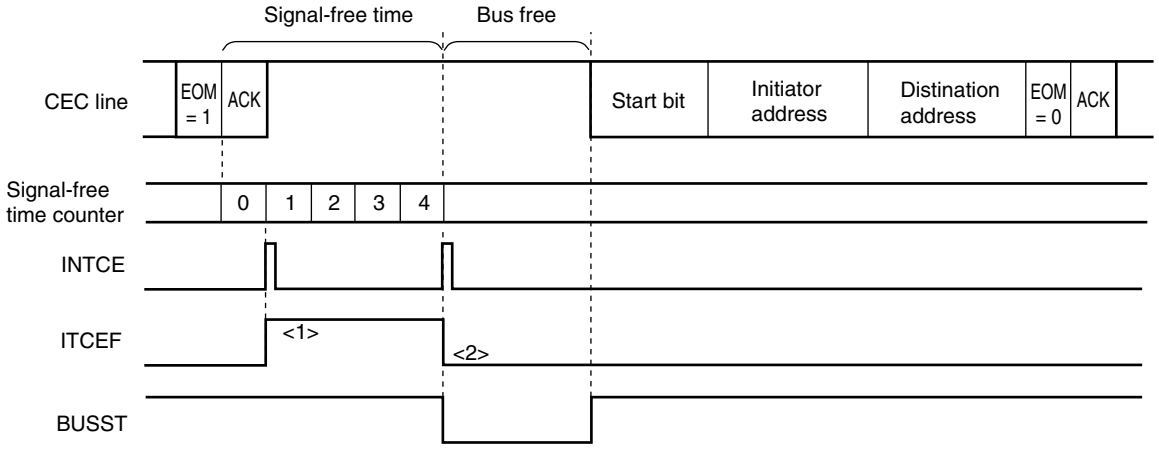
Symbol	7	6	5	4	3	2	1	0
CECS	SFTST	0	0	ITCEF	EOMF	TXST	BUSST	ADRF

BUSST	Bus busy detection flag
0	Bus-free state BUSST is reset to 0 in the following cases. • BUSST is reset to 0 regardless of the bus state when CECE = 0. • When the signal-free time set by the SFT1 and SFT0 bits has elapsed after the communication has ended (see **Figure 13-10**)
1	Bus busy state BUSST is set to 1 in the following cases. • When a fall of the CEC bus is detected (see **Figure 13-11**) • When CECE = 1 is set during a reception or standby period (see **Figure 13-12**)

ADRF	Address match detection flag
0	During a communication between other stations, while communication is stopped, or while the local station is transmitting ADRF is reset to 0 in the following cases. • When CECE = 0 • When reception is completed
1	During local reception ADRF is set to 1 in the following cases. • When the local address and reception destination address match • When a broadcast address is received

Figure 13-9. Checking INTCE Generation Source Using ITCEF When CESEL0 and CESEL1 Are 0

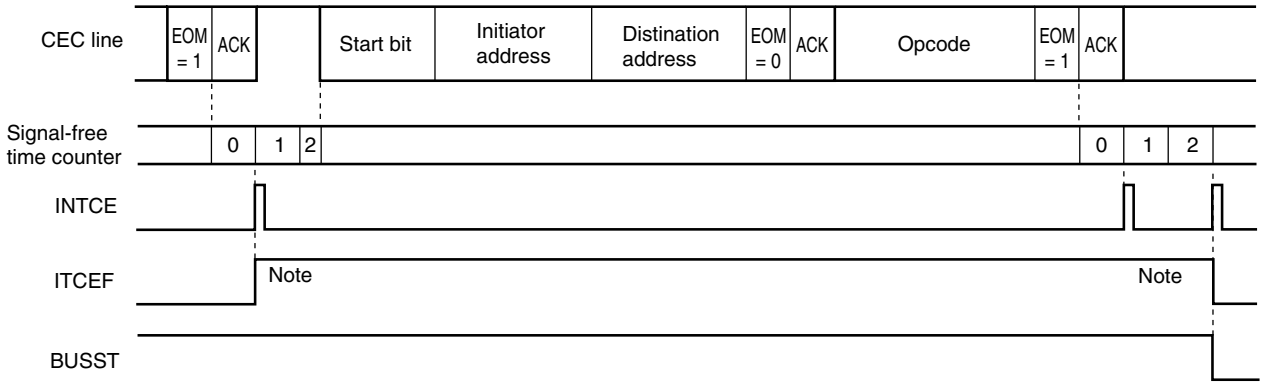
(1) Reception started after the signal-free time was counted



If ITCEF is 1, INTCE is generated after ACK is received (<1>).

If ITCEF is 0, INTCE is generated after the signal-free time is counted (<2>).

(2) Reception started during the signal-free time



Note ITCEF is 1 until INTCE is generated after the signal-free time is counted.

Figure 13-10. Timing When Signal-Free Time Set by SFT1 and SFT0 Bits Has Elapsed After Communication End

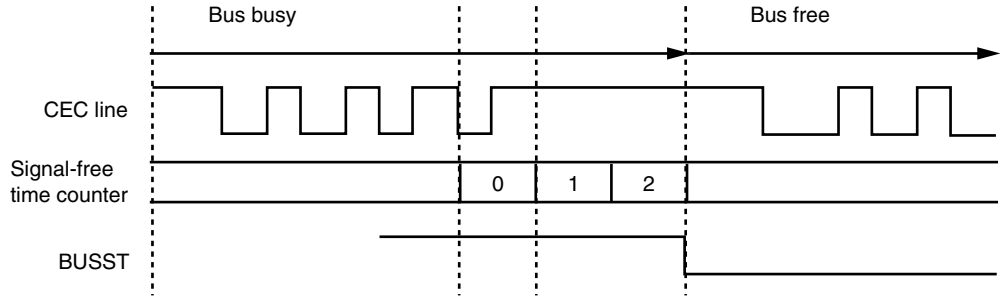


Figure 13-11. CEC Line Fall Detection Timing

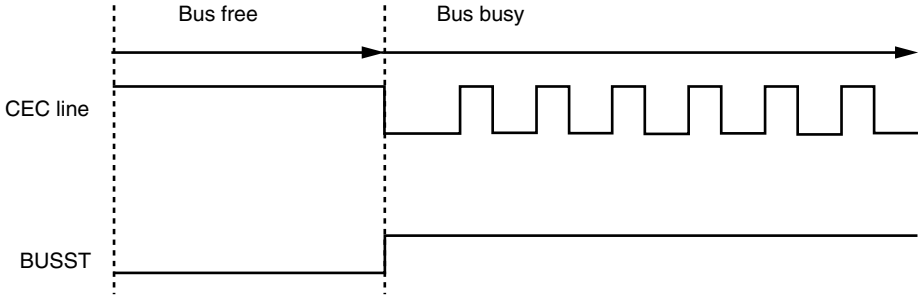
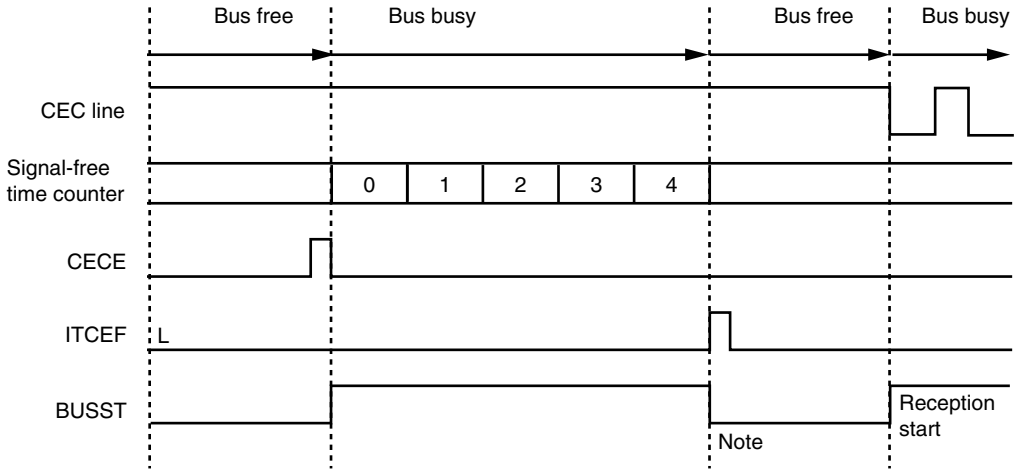


Figure 13-12. Timing When CECE = 1 Is Set During Reception or Standby Period



Note Transmission errors are not detected while transmitting the start bit and ACK bit.

(8) CEC communication error status register (CECES)

CECES indicates whether a bus lock error, arbitration loss, transmission error, timing error, ACK error, underrun error, or overrun error has detected.

CECES can be read by an 8-bit memory manipulation instruction.

Clearing CECE to 0 or generating a reset signal clears this register to 00H.

Figure 13-13. Format of CEC Communication Error Status Register (CECES) (1/2)

Address: FFF7AH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
CECES	0	BLERR	AERR	TXERR	TERR	ACKERR	UERR	OERR

BLERR	Bus lock error detection flag
0	No bus lock error has occurred. BLERR is cleared to 0 in the following case. • When BLCTRG = 1 is set
1	A bus lock error has occurred. BLERR is set to 1 in the following case. • When the CEC reception signal is fixed to low or high level midway through a frame An error occurs if the next falling edge is not input for a time 2.5 times the 1-data bit width set by the NOMP register after the falling edge of the CEC reception signal.

AERR	Arbitration loss detection flag
0	No arbitration loss has occurred or the communication is being stopped. AERR is reset to 0 in the following case. • When ACTRG = 1
1	Arbitration loss AERR is set to 1 in the following case. • When an arbitration loss occurs between start bit transmission and source address transmission

TXERR	Transmission error detection flag[Note]
0	No transmission error has occurred. TXERR is reset to 0 in the following case. • When TXCTRG = 1
1	A transmission error has occurred. TXERR is set to 1 in the following case. • When the logic of the transmit data and receive data are compared and do not match when the initiator is operating

Note Transmission errors are not detected while transmitting the start bit and ACK bit.

Figure 13-13. Format of CEC Communication Error Status Register (CECES) (2/2)

Address: FFF7AH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
CECES	0	BLERR	AERR	TXERR	TERR	ACKERR	UERR	OERR

TERR	Timing error detection flag
0	No timing error has occurred. TERR is reset to 0 in the following case. • When TCTRG = 1
1	A timing error has occurred. TERR is set to 1 in the following case. • When a violation is detected in the timing check of the received data

ACKERR	ACK error detection flag
0	No ACK error has occurred. ACKERR is reset to 0 in the following case. • When ACKTRG = 1
1	An ACK error has occurred. ACKERR is set to 1 in the following cases. • When a NACK (logical 1) is received during a direct address communication • When an ACK (logical 0) is received during a broadcast communication • When an NACK (logical 1) is received during a logical address allocation transmission

UERR	Underrun error detection flag
0	No underrun error has occurred. UERR is reset to 0 in the following case. • When UECTRG = 1
1	An underrun error has occurred. UERR is set to 1 in the following case. • When transmit data is not written to the transmission buffer register (CTXD) after a data interrupt (INTDA) is generated and before the next interrupt (INTDA) is generated

OERR	Overrun error detection flag
0	No overrun error has occurred. OERR is reset to 0 in the following case. • When OCTRG = 1
1	An overrun error has occurred. OERR is set to 1 in the following case. • When the next receive operation is completed before the receive data stored in the reception buffer register (CRXD) is read

(9) CEC communication error flag clear trigger register (CECFC)

CECFC clears error flags written to the communication error status register (CECES). Only the set bits can be cleared by setting “1” to each flag.

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

Reset signal generation clears this register to 00H.

Figure 13-14. Format of CEC Communication Error Flag Clear Trigger Register (CECFC)

Address: FFF7CH After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
CECFC	0	BLCTRG	ACTRG	TXCTRG	TCTRG	ACKCTRG	UCTRG	OCTRG

BLCTRG	Bus lock error clear trigger
0	Does not clear bus lock error flag.
1	Clears bus lock error flag.

ACTRG	Arbitration loss flag clear trigger
0	Does not clear arbitration loss flag.
1	Clears arbitration loss flag.

TXCTRG	Transmission error flag clear trigger
0	Does not clear transmission error flag.
1	Clears transmission error flag.

TCTRG	Timing error flag clear trigger
0	Does not clear timing error flag.
1	Clears timing error flag.

ACKCTRG	ACK error flag clear trigger
0	Does not clear ACK error flag.
1	Clears ACK error flag.

UCTRG	Underrun error flag clear trigger
0	Does not clear underrun error flag.
1	Clears underrun error flag.

OCTRG	Overrun error flag clear trigger
0	Does not clear overrun error flag.
1	Clears overrun error flag.

(10) Port function register 6 (PF6)

PF6 sets whether to use the P62/CECIO pin of port 6 in I/O port mode or CECIO mode.

A normal input buffer or a CEC input buffer can be specified by setting the PF6 register and port input mode register 6 (PIM6).

Whether to use a diode connection can be set by setting the PF6 register and pull-up resistor option register 6 (PU6).

PF6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13-15. Format of Port Function Register 6 (PF6)

Address: F0076H After reset: 00 R/W

Symbol	7	6	5	4	3	<2>	1	0
PF6	0	0	0	0	0	PF62	0	0

PF62	P62 pin operating mode selection
0	I/O port mode
1	CECIO mode

Figure 13-16. Relationship Between PF6 Register and PIM6 and PU6 Registers

PF62	PIM62	Input buffer selection
0	0	Normal input buffer
0	1	
1	0	
1	1	CEC input buffer

PF62	PU62	Input buffer selection
0	0	Does not connect pull-up resistor + diode.
0	1	
1	0	
1	1	Connects pull-up resistor + diode.

Cautions 1. Using the CECIO and CECIN/CECOUT pins at the same time is prohibited.
When PF62 = 1, do not set PF110 and PF111 to 1.
When PF110 = 1 or PF111 = 1, do not set PF62 to 1.
2. When PF62 = 1 is set, set PM62 = 0 and P62 = 1.

(11) Port function register 7 (PF7)

PF11 sets whether to use the P110/CECIN pin of port 11 in I/O port mode or CECIN mode.

PF11 sets whether to use the P111/CECOUT pin of port 11 in I/O port mode or CECOUT mode.

PF11 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 Port Function Register 11 (PF11)

Address: F007BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PF11	0	0	0	0	0	0	PF111	PF110

PF110	P110 pin operating mode selection
0	I/O port mode
1	CECIN mode

PF111	P111 pin operating mode selection
0	I/O port mode
1	CECOUT mode

Caution Using the CECIO and CECIN/CECOUT pins at the same time is prohibited.
When PF62 = 1, do not set PF110 and PF111 to 1.
When PF110 = 1 or PF111 = 1, do not set PF62 to 1.

13.5 Start Bit and Data Bit Registers

13.5.1 Specifying low-level width and bit width of CEC transmission data

The low-level width and bit width of the transmit data are set by using the following registers.

(1) CEC transmission start bit low width setting register (STATL)

STATL is a 9-bit register that sets the low-level width of the start bit during a transmission.

STATL can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-18. Format of CEC Transmission Start Bit Low Width Setting Register (STATL)

Address: F0306H, FA07H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATL	0	0	0	0	0	0	0	STATL8 to STATL0								

Remark Low-level width = (Set values of STATL8 to STATL0 + 1) × Clock cycle of f_{CEC}

(2) CEC transmission start bit width setting register (STATB)

STATB is a 9-bit register that sets the bit width of the start bit during a transmission.

STATB can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-19. Format of CEC Transmission Start Bit Width Setting Register (STATB)

Address: F0304H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATB	0	0	0	0	0	0	0	STATB8 to STATB0								

Remark Bit width = (Set values of STATB8 to STATB0 + 1) × Clock cycle of f_{CEC}

(3) CEC transmission logical 0 low width setting register (LGC0L)

LGC0L is a 9-bit register that sets the low-level width of logical 0 during a transmission.

LGC0L can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-20. Format of CEC Transmission Logical 0 Low Width Setting Register (LGC0L)

Address: F0308H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LGC0L	0	0	0	0	0	0	0	LGC0L8 to LGC0L0								

Remark Low-level width = (Set values of LGC0L8 to LGC0L0 + 1) × Clock cycle of f_{CEC}

(4) CEC transmission logical 1 low width setting register (LGC1L)

LGC1L is a 9-bit register that sets the low-level width of logical 1 during a transmission.

LGC1L can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-21. Format of CEC Transmission Logical 1 Low Width Setting Register (LGC1L)

Address: F030AH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LGC1L	0	0	0	0	0	0	0	LGC1L8 to LGC1L0								

Remark Low-level width = (Set values of LGC1L8 to LGC1L0 + 1) × Clock cycle of f_{CEC}

(5) CEC transmission data bit width setting register (DATB)

DATB is a 9-bit register that sets the bit width of the data bit during a transmission.

DATB can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-22. Format of CEC Transmission Data Bit Width Setting Register (DATB)

Address: F030CH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATB	0	0	0	0	0	0	0	DATB8 to DATB0								

Remark 1-data bit width = (Set values of DATB8 to DATB0 + 1) × Clock cycle of f_{CEC}

13.5.2 Checking CEC reception data timing

The timing at which errors in the low-level width and bit width of the receive data are judged is set by the following registers.

(1) CEC reception start bit minimum low width setting register (STATLL)

STATLL is a 9-bit register that detects the minimum value of the low-level width of the start bit during a reception.

STATLL can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-23. Format of CEC Reception Start Bit Minimum Low Width Setting Register (STATLL)

Address: F0310H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATLL	0	0	0	0	0	0	0	STATLL8 to STATLL0								

Remark Low-level width = (Set values of STATLL8 to STATLL0 + 1) × Clock cycle of f_{CEC}

(2) CEC reception start bit maximum low width setting register (STATLH)

STATLH is a 9-bit register that detects the maximum value of the low-level width of the start bit during a reception.

STATLH can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-24. Format of CEC Reception Start Bit Maximum Low Width Setting Register (STATLH)

Address: F0312H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATLH	0	0	0	0	0	0	0	STATLH8 to STATLH0								

Remark Low-level width = (Set values of STATLH8 to STATLH0 + 1) × Clock cycle of f_{CEC}

(3) CEC reception start bit minimum bit width setting register (STATBL)

STATBL is a 9-bit register that sets the minimum value of the bit width of the start bit during a reception.

STATBL can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-25. Format of CEC Reception Start Bit Minimum Bit Width Setting Register (STATBL)

Address: F0314H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATBL	0	0	0	0	0	0	0	STATBL8 to STATBL0								

Remark Bit width = (Set values of STATBL8 to STATBL0 + 1) × Clock cycle of f_{CEC}

(4) CEC reception start bit maximum bit width setting register (STATBH)

STATBH sets the maximum value of the bit width of the start bit during a reception.

STATBH can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-26. Format of CEC Reception Start Bit Maximum Bit Width Setting Register (STATBH)

Address: F0316H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STATBH	0	0	0	0	0	0	0	STATBH8 to STATBH0								

Remark Bit width = (Set values of STATBH8 to STATBH0 + 1) × Clock cycle of f_{CEC}

(5) CEC reception logical 0 minimum low width setting register (LGC0LL)

LGC0LL is a 9-bit register that sets the minimum value of the low-level width of logical 0 during a reception.

LGC0LL can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-27. Format of CEC Reception Logical 0 Minimum Low Width Setting Register (LGC0LL)

Address: F0318H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LGC0LL	0	0	0	0	0	0	0	LGC0LL8 to LGC0LL0								

Remark Low-level width = (Set values of LGC0LL8 to LGC0LL0 + 1) × Clock cycle of f_{CEC}

(6) CEC reception logical 0 maximum low width setting register (LGC0LH)

LGC0LH is a 9-bit register that sets the maximum value of the low-level width of logical 0 during a reception.

LGC0LH can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-28. Format of CEC Reception Logical 0 Maximum Low Width Setting Register (LGC0LH)

Address: F031AH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LGC0LH	0	0	0	0	0	0	0	LGC0LH8 to LGC0LH0								

Remark Low-level width = (Set values of LGC0LH8 to LGC0LH0 + 1) × Clock cycle of f_{CEC}

(7) CEC reception logical 1 minimum low width setting register (LGC1LL)

LGC1LL is a 9-bit register that sets the minimum value of the low-level width of logical 1 during a reception.

LGC1LL can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-29. Format of CEC Reception Logical 1 Minimum Low Width Setting Register (LGC1LL)

Address: F031CH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LGC1LL	0	0	0	0	0	0	0	LGC1LL8 to LGC1LL0								

Remark Low-level width = (Set values of LGC1LL8 to LGC1LL0 + 1) × Clock cycle of f_{CEC}

(8) CEC reception logical 1 maximum low width setting register (LGC1LH)

LGC1LH is a 9-bit register that sets the maximum value of the low-level width of logical 1 during a reception.

LGC1LH can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-30. Format of CEC Reception Logical 1 Maximum Low Width Setting Register (LGC1LH)

Address: F031EH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LGC1LH	0	0	0	0	0	0	0	LGC1LH8 to LGC1LH0								

Remark Low-level width = (Set values of LGC1LH8 to LGC1LH0 + 1) × Clock cycle of f_{CEC}

(9) CEC reception data bit minimum bit width setting register (DATBL)

DATBL is a 9-bit register that sets the minimum value of the bit width of the data bit during a reception.

DATBL can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-31. Format of CEC Reception Data Bit Minimum Bit Width Setting Register (DATBL)

Address: F0320H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATBL	0	0	0	0	0	0	0	DATBL8 to DATBL0								

Remark Bit width = (Set values of DATBL8 to DATBL0 + 1) × Clock cycle of f_{CEC}

(10) CEC reception data bit maximum bit width setting register (DATBH)

DATBH is a 9-bit register that sets the maximum value of the bit width of the data bit during a reception.

DATBH can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-32. Format of CEC Reception Data Bit Maximum Bit Width Setting Register (DATBH)

Address: F0322H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATBH	0	0	0	0	0	0	0	DATBH8 to DATBH0								

Remark Bit width = (Set values of DATBH8 to DATBH0 + 1) × Clock cycle of f_{CEC}

13.5.3 Determining CEC reception data logic (1 or 0)

The timing at which receive data is judged to be 1 or 0 is set by the following register.

(1) CEC reception data sampling time setting register (NOMT)

NOMT is a 9-bit register that determines the sampling time of data during a reception.

NOMT can be set by a 16-bit memory manipulation instruction and rewritten only when CECE = 0.

Reset signal generation clears this register to 0000H.

Figure 13-33. Format of CEC Reception Data Sampling Time Setting Register (NOMT)

Address: F030EH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOMT	0	0	0	0	0	0	0	NOMT8 to NOMT0								

Remark Sampling time = (Set values of NOMT8 to NOMT0 + 1) × Clock cycle of f_{CEC}

Set this register within a period of LGC1LH < NOMT < LGC0LL.

13.5.4 Specifying bit width for detecting errors, signal-free time, and bus locking

The bit width used when detecting errors, the signal-free time, and bus locking is specified using the following register:

(1) CEC data bit reference width setting register (NOMP)

NOMP is a 9-bit register for specifying the bit width.

The bit width is used when counting the number of bits when detecting errors, the signal-free time, and bus locking.

NOMP can be set by a 16-bit memory manipulation instruction. It can be rewritten only if CECE is 0.

Reset signal generation clears this register to 0000H.

Figure 13-34. Format of CEC Data Bit Reference Wide Setting Register (NOMP)

Address: F0324H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOMP	0	0	0	0	0	0	0	NOMT8 to NOMT0								

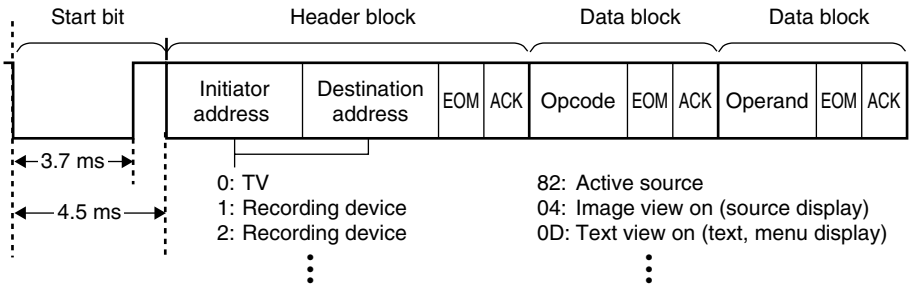
Remark Bit width = (Set values of NOMP8 to NOMP0 + 1) × Clock cycle of f_{CEC}

13.6 Operation of CEC Transmission/Reception Circuit

13.6.1 CEC transmission/reception data format

Figure 13-35 shows the basic CEC communication format. A CEC data frame consists of a start bit, a header block, data block 1 (opcode), and data block 2 (operand). The three blocks other than the start bit are configured of 10 bits.

Figure 13-35. Format Example



Start bit: Bit indicating the start of a message

Header block (1 block): Block indicating the source and destination addresses. Arbitration for the source address is performed at this block and the initiator with the smaller address obtains the transmission right.

Data block (blocks 0 to 15): Block consisting of an opcode and operand. The data length of the operand is determined by the opcode.

13.6.2 Communication types

CEC transmission/reception takes place in the state of a direct address message or broadcast address message. In a CEC communication, the transmitting side transmits a start bit and message (data) and the receiving side receives the message and returns a desired acknowledge signal to the transmitting side. CEC transmission/reception is configured of a start bit and a data bit and performs all transmissions/receptions of CEC.

Remark Direct address message: When the receiving party of a transmission is the local device
Broadcast address message: When the receiving party of a transmission is a device other than the local device

13.6.3 Bit timing

Figure 13-36 shows an example of the pulse format of a start bit. Whether the start bit is valid is judged at the low-level period (a) and bit period (b).

Figure 13-36. Start Bit Format

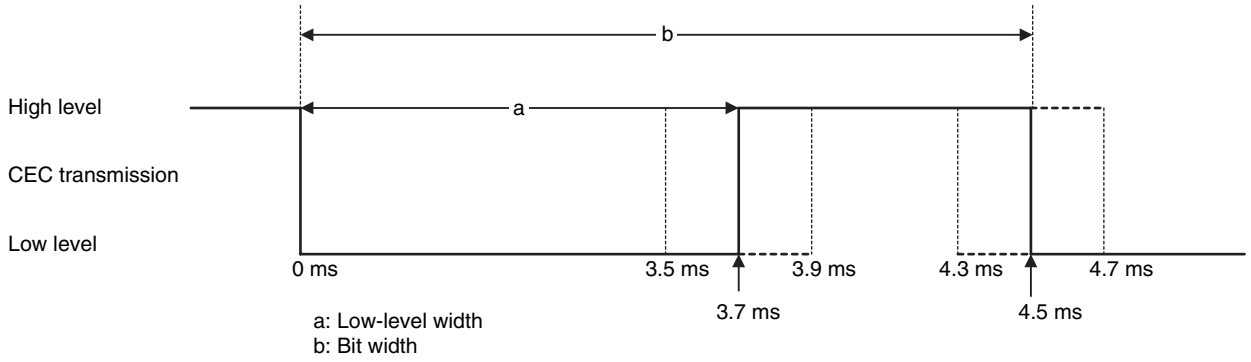
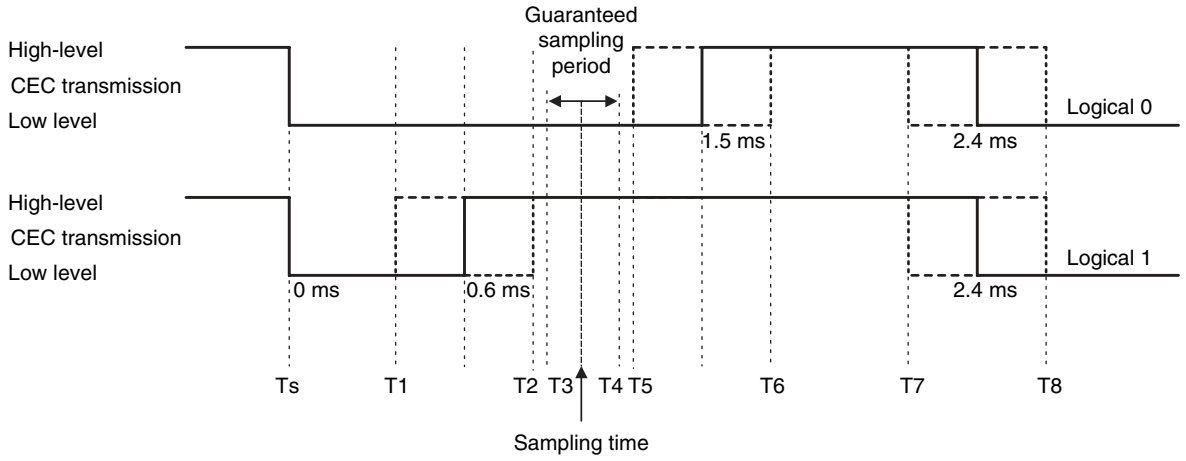


Figure 13-37 shows an example of the pulse format of a data bit timing. A data bit is sampled at a sampling timing set with the CEC reception data sampling time setting register (NOMT). If the result of sampling is low level, the pulse format is logical 0. If the result of sampling is high level, the pulse format is logical 1. The last change from high to low level of a data bit is the start of the next data bit. Consequently, the last data bit remains at high level.

Figure 13-37. Data Bit Format



Ts	0 ms	The bit start event
T1	0.4 ms	The earliest time for a low-high transition when indicating a logical 1
T2	0.8 ms	The latest time for a low-high transition when indicating a logical 1
T3	0.85 ms	The earliest time it is safe to sample the signal line to determine its state
T4	1.25 ms	The latest time it is safe to sample the signal line to determine its state
T5	1.3 ms	The earliest time a device is permitted return to a high impedance state(logical 0)
T6	1.7 ms	The latest time a device is permitted return to a high impedance state(logical 0)
T7	2.05 ms	The earliest time for the start of a following bit
	2.4 ms	The nominal data bit period
T8	2.75 ms	The latest time for the start of a following bit

13.6.4 Header/data block

All data blocks are configured of 10 bits and have the same structure. Figure 13-38 shows the configuration of header and data blocks. An information bit has a different meaning for a header block and data block, and indicates the data, opcode, and address. EOM (End of Message) and ACK (Acknowledge) are control bits and have the same meanings for a header block and data block.

Figure 13-38. Header Block and Data Block Format

Header/data block									
7	6	5	4	3	2	1	0	–	–
Information bit								EOM	ACK

A header block consists of an initiator logical address, destination logical address, EOM (End of Message), and ACK (Acknowledge). The EOM of a header block is used for “ping” with another device (checking whether the power of another device is turned on). “ping” can be checked by setting EOM = 1 and transmitting only the header block (transmitting a message without data blocks). In the case of direct address transmission, the power of the device to which the header block is transmitted is turned on if an ACK is returned.

13.6.5 EOM (End of Message)

An EOM indicates whether the transmitted block is the last block of a message. It is added to an information bit and output.

Caution EOM bit = 0: When one block or multiple blocks follow
EOM bit = 1: When the transmitted block is the last block

13.6.6 ACK (Acknowledge)

The meaning of an ACK depends on whether the receiving party of a transmission is a direct address message or broadcast message. The result of comparing the received data and CEC line data is transmitted to the transmitting side as an ACK or NACK.

Direct communication: An ACK is transmitted if the comparison result is normal.
Broadcast communication: An NACK is transmitted if the comparison result is normal.

Cautions 1. ACK = 0 is the normal value for a direct address message.

<1> If no error exists in the header block and the local address is used, the ACK bit is 0.
<2> If no error exists in the data blocks, the ACK bit is 0.
<3> If an error exists in the header block or another address is used, the ACK bit is 1.
<4> If an error exists in the data blocks, the ACK bit is 1.

An NACK (ACK = 1) is the normal value for a broadcast message.
<1> If one or more followers have abandoned the message, the ACK bit is 0.
<2> If all followers have not abandoned the message, the ACK bit is 1.

2. The initiator always outputs logical 1 at the ACK bit timing. Consequently, a follower determines the logic level of the ACK bit.

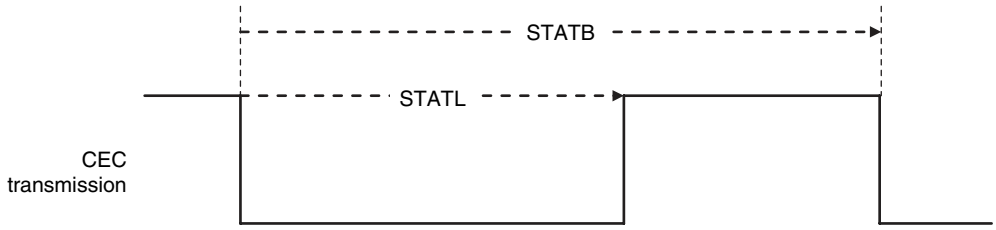
13.7 CEC Communication Functions

13.7.1 Communication bit width adjustment function

This function can be used to set the low-level width and bit width of the start bit and data bit during a transmission.
The relationships between the various width setting registers (see **13.5.1**) and the bit timing are shown below.

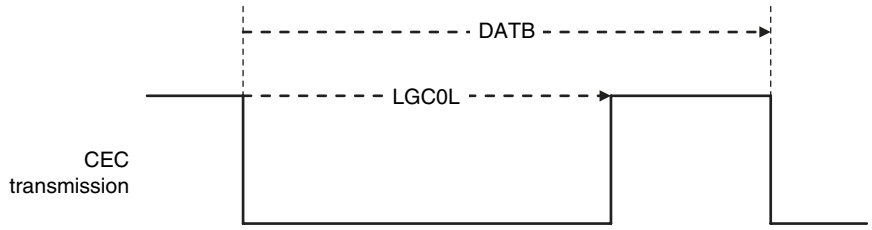
<Start bit>
The STATL register is used to set the low-level width and the STATB register is used to set the bit width of the start bit.

Figure 13-39. Start Bit Output Waveform



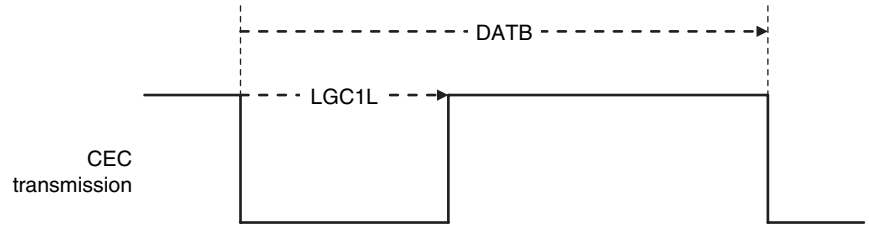
<Data bit (logical 0)>
The LGC0L register is used to set the low-level width and the DATB register is used to set the bit width of the data bit of logical 0.

Figure 13-40. Data Bit (Logical 0) Output Waveform



<Data bit (logical 1)>
The LGC1L register is used to set the low-level width and the DATB register is used to set the bit width of the data bit of logical 1.

Figure 13-41. Data Bit (Logical 1) Output Waveform



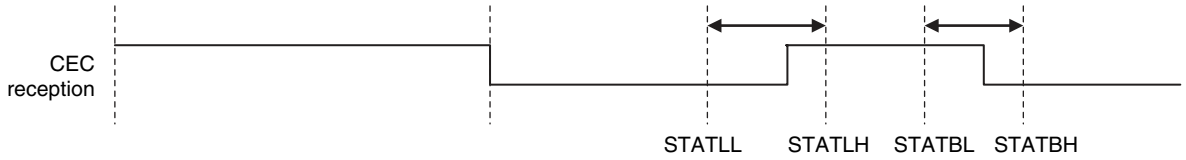
13.7.2 Receive bit timing check function

The CEC transmission/reception circuit has a timing check function that judges whether the low-level width and bit width of the start bit and data bit during a reception are within the set range. The timing check time can be set by using the various timing judgment registers (see **13.5.2**). The relationships between the timing judgment registers and bit timing are shown below.

<Start bit>

The STATLL register is used to set the minimum low-level width and the STATLH register is used to set the maximum low-level width of the start bit. The STATBL register is used to set the minimum value and the STATBH register is used to set the maximum value of the start bit width.

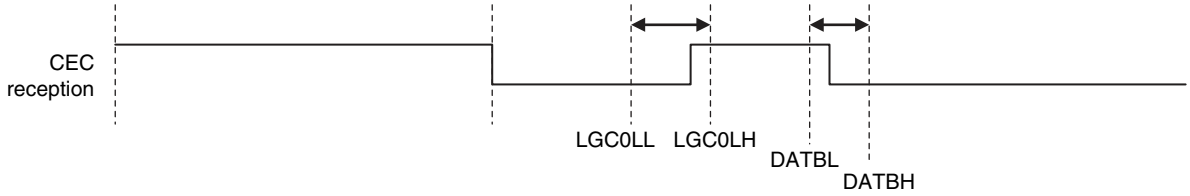
Figure 13-42. Start Bit Reception Timing



<Data bit (logical 0)>

The LGC0LL register is used to set the minimum low-level width and the LGC0LH register is used to set the maximum low-level width of the data bit (logical 0). The DATBL register is used to set the minimum value and the DATBH register is used to set the maximum value of the bit width.

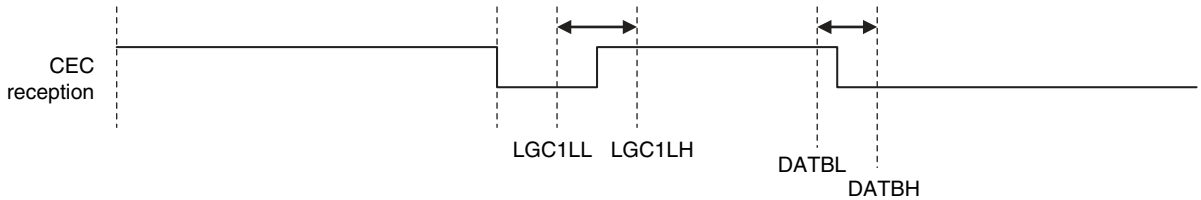
Figure 13-43. Data Bit (Logical 0) Reception Timing



<Data bit (logical 1)>

The LGC1LL register is used to set the minimum low-level width and the LGC1LH register is used to set the maximum low-level width of the data bit (logical 1). The DATBL register is used to set the minimum value and the DATBH register is used to set the maximum value of the bit width.

Figure 13-44. Data Bit (Logical 1) Reception Timing

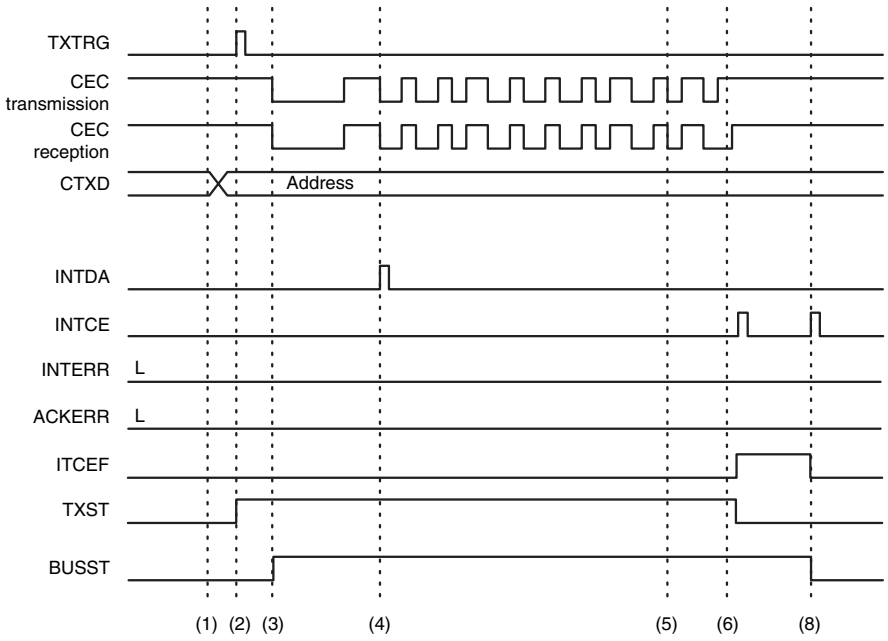


13.7.3 Initial CEC communication settings

The initial CEC communication setting flow is explained below. The logical address acquisition flow is executed by setting the various control registers and using direct address transmission after a reset. In a logical address acquisition transmission, EOM = 1 is set because the same address is set for the source and destination addresses and only the header block is transmitted. Furthermore, to prevent a false address match from occurring before the local address is determined, CECRXEN = 0 must be set until the CADR setting. Figure 13-45 shows the logical allocation timing diagram and Figure 13-46 the operation procedure and an explanation of the operation.

Figure 13-45. Logical Allocation (CECSEL0, CECSEL1 = 0, 0)

- The address is used by another device



- The address is not used by another device

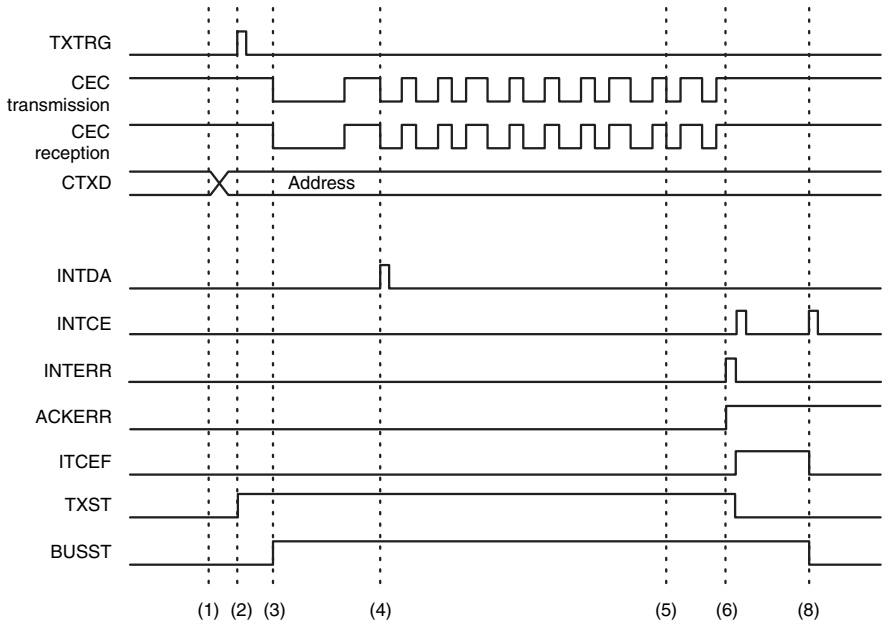


Figure 13-46. Initial CEC Communication Setting Procedure (1/2)

	Software Manipulation	Hardware State
Initial CEC setting	[Operating clock (f_{CEC}) setting] Set CCL2 to CCL0. [Reception rejection control setting] Set CECRXEN to 0. [Setting for reporting address mismatch] Set CINTMK. [Noise elimination selection] Set CDFC. (Specify whether to use the noise filter.) [Start bit low-level/bit width setting] Set the low/bit width to STATL/STATB. [Logical 0/1 low-level/bit width setting] Set the low/bit width to LGC0L/LGC1L/DATB. [Sampling time setting] Specify the time to sample the received data for NOMT. [Bit width setting] Specify the bit width for NOMP. [Timing check register settings] Set the low-level width timing check time of the start bit to STATLL/STATLH. Set the bit width timing check time of the start bit to STATBL/STATBH. Set the low-level width timing check time of the data bit (logical 0) to LGC0LL/LGC0LH. Set the low-level width timing check time of the data bit (logical 1) to LGC1LL/LGC1LH. Set the bit width timing check time of the data bit to DATBL/DATBH. [Bus lock detection setting] Set up BLERRD. (Select whether to detect bus locking.) [Start bit timing error detection setting] Set STERRD. (Select whether to detect timing errors of the start bit.) [Communication complete interrupt setting] Set CESEL1 and CESEL0. [Signal-free time setting] Set SFT1 and SFT0. (Set the signal-free time detection time.) [Operating clock (f_{CEC}) supply] Set CECE to 1. →	The operating clock is stopped. The operating clock (f_{CEC}) is supplied. Transmission can be performed. The signal-free time is started. BUSST becomes 1. BUSST becomes 0 and the communication standby state is entered after counting up to the set values of SFT1 and SFT0.

Figure 13-46. Initial CEC Communication Setting Procedure (2/2)

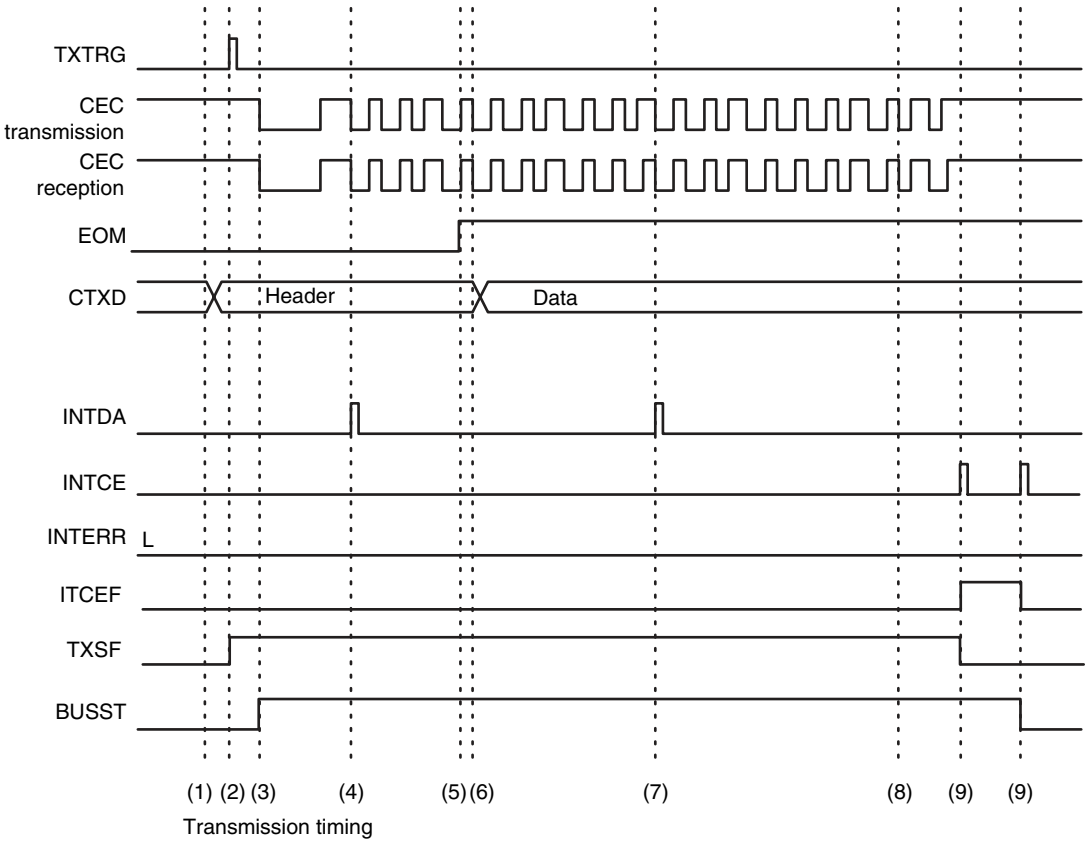
	Software Manipulation	Hardware State
Logical address allocation	[EOM setting] Set EOM to 1. [Transmit data setting] (1) Set the transmit data (logical address) to CTXD. [Bus-free state check] Check that BUSST is 0. [Starting transmit operation] (2) Set TXTRG to 1. →	Transmission is started. The start bit is output (3).
	Do not write the next data, because only the header block is transmitted. ←	The values set to the CTXD register are sequentially output at the same time as INTDA is output when the header block output is started (4).
		1 is output from the EOM bit (5).
	[Local address setting]	
		The ACK bit is received.
	• ACK Change the transmitted address and then retransmit the address, because it is used by another station. ←	When logical 0 is received, INTERR is not output and the ACKERR flag is not set (6).
	• NACK Use the transmitted address as the local address, because the transmitted address is not used by another station (CADR setting). ←	When logical 1 is received, INTERR is output and the ACKERR flag is set (7).
		INTCE is output according to the CESEL1, CESEL0, SFT1, and SFT0 bit settings (8).
	[Reception rejection control setting] Set CECRXEN to 1. →	The communication standby state is entered.

13.7.4 CEC transmission

CEC transmission is explained below. This section assumes that the settings in **13.7.3 Initial CEC communication settings** are completed.

A CEC transmission performs a receive operation even during transmission and performs an arbitration check, a data check, and a timing check. The value of the receive data register (CRXD) during a transmit operation, however, is not guaranteed.

Figure 13-47. Basic Transmission Timing (Direct Address Transmission) (CESEL0, CESEL1 = 0, 0)



Caution A new start bit detected during transmission (the time from when a start bit is detected until the EOM of the last frame is received) is ignored and detected as a timing error. Communication is not restarted.

(1) CEC transmission manipulation procedure

Figure 13-48. CEC Transmission Manipulation Procedure

	Software Manipulation	Hardware State
CEC transmit operation	[Signal-free time setting] Set SFT1 and SFT0. (Set the signal-free time detection time.) [EOM setting] (1) Set EOM (EOM = 0). [Transmit data setting] (1) Set the transmit data to CTXD. [Bus-free state check] Check that BUSST is 0. [Starting transmit operation] (2) Set TXTRG to 1.	Transmission is started. The start bit is output (3).
	[EOM setting] (5) Set the EOM of the next frame (EOM = 1) before the next frame starts (7).	The values set to the CTXD register are sequentially output at the same time as INTDA is output when the header block output is started (4).
	[Transmit data setting] (6) Set the transmit data to CTXD.	Outputting the data of the second frame is started (7). 1 is output at the EOM bit position because the last frame is reached (8). INTCE is output according to the CESEL1, CESEL0, SFT1, and SFT0 bit settings (9). The communication standby state is entered.

(2) Broadcast transmission

When FH is set to the destination address of the header block transmit data (CTXD), the hardware recognizes the current transmission as a broadcast transmission and operates. Normally, the communication is judged as being successfully performed when logical 0 is received at the ACK bit timing; however, in broadcast communication, the communication is judged as being successfully performed when logical 1 is received at the ACK bit timing.

The hardware judges whether the communication is a direct communication or broadcast communication by looking at the transmit data of the header block, and it automatically determines whether the reception of logical 0 or logical 1 has been successfully performed.

(3) CEC transmission interrupt

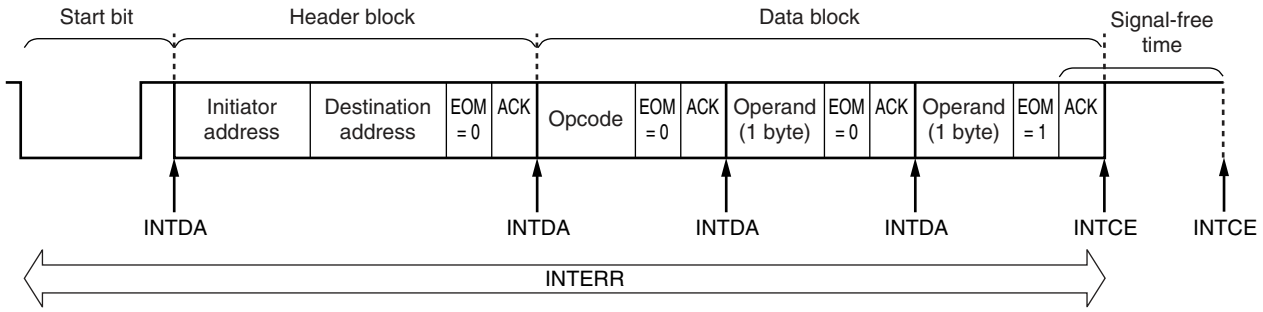
The hardware has three interrupt functions, namely a data interrupt (INTDA), a communication complete interrupt (INTCE), and an error interrupt (INTERR).

A data interrupt (INTDA) occurs at the start of each block.

A communication complete interrupt (INTCE) can be generated if ACK reception for a data block for which EOM is set to 1 ends, if the signal-free time specified using SFT1 and SFT0 elapses, or if both conditions occur, depending on the settings of CESEL1 and CESEL0.

An error interrupt (INTERR) is generated if a timing error, ACK error, underrun error, transmission error, or bus lock error is detected during any period of time during communication.

Figure 13-49. Interrupt Generation Timing



Caution If the falling edge of the CEC line is detected when receiving the ACK bit by setting EOM to 1 (before receiving the ACK bit ends), an irregular operation is performed as shown in Table 13-2 according to that timing.

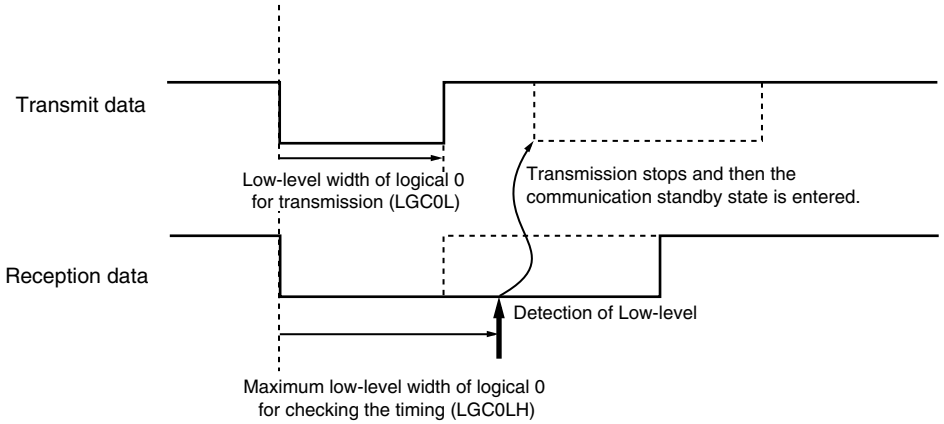
Table 13-2. Operation If Falling Edge of CEC Line Is Detected before Receiving ACK Bit Ends

CEC Line Falling Timing	Values Specified for CESEL1 and CESEL0 Bits of CECCTL1 Register	INTCE Generation	Handling of ACK Bit	Operation after CEC Line Falls
After the minimum data bit value (DATBL ≤ counter)	CESEL1 and CESEL0 are 0 and 0, or 0 and 1, respectively.	INTCE is generated once when the CEC line falls.	Handling the ACK bit is enabled because it has the correct width. (ACK or NACK is correctly determined.)	The start of the next communication is recognized and then determining whether to receive the start bit starts.
	CESEL1 and CESEL0 are 1 and 0, respectively.	INTCE is not generated.		
Before the minimum data bit value (counter < DATBL)	CESEL1 and CESEL0 are 0 and 0, or 0 and 1, respectively.	INTCE is generated once when the CEC line falls.	ACK cannot be correctly determined because it has the incorrect width. (If ACKTEN is set to 1, a timing error occurs.)	
	CESEL1 and CESEL0 are 1 and 0, respectively.	INTCE is not generated.		

(4) Receiving error handling pulse

If the received data is at low level when the maximum low-level width of logical 0 is reached while the initiator runs, an error handling pulse is judged to be received, a timing error occurs, transmission stops, and then the communication standby state is entered.

Figure 13-50. Interrupt Generation Timing



13.7.5 CEC reception

CEC reception is explained below. This section assumes that the settings in **13.7.3 Initial CEC communication settings** are completed. During reception, the data is received at the sampling timing set by the CEC reception data sampling time setting register (NOMT) and stored in CRXD.

The receive operation differs depending on the CECRXEN bit setting value, CINTMK bit setting value, communication type (direct address communication or broadcast communication), and whether the reception address and local address match.

The correspondences between various conditions and the operations are shown in the following table.

Table 13-3. Corresponding Operation During CEC Reception

CEC RXEN	Communication Type	Address Match/ Mismatch	CINTMK Bit	BUSST Operation	INTDA Output	INTCE Output	INTERR Output	Error Flag Operation	Error Detection (Other than Short Bit Width Detection)	Error Detection (Short Bit Width Detection)	Error Handling Output	Bus Lock Detection Note 1	ACK/NACK Output	Signal-Free Time Count
0	–	–	–	√	×	×	×	×	×	×	×	√Note 2	×	×
1	Start bit	–	–	√	×	√Note 3	×	×	√Note 4	√Note 4	×	√	×	√
	Header	Mismatch	0	√	×	√Note 3	√	√	√	√	√	√	√	√
			1	√	√	√	√	√	√	√	√	√	√	√
		Match	–	√	√	√	√	√	√	√	√	√	√	√
	Direct (data)	Mismatch	0	√	×	×	×	×	×	√	√	√	×	√
			1	√	√	√	√	√	√	√	√	√	×	√
		Match	–	√	√	√	√	√	√	√	√	√	√	√
	Broadcast (data)	–	–	√	√	√	√	√	√	√	√	√	√	√

Notes 1. Bus lock errors are detected by setting BLERRD.

2. A bus lock error is detected but a flag is not set.

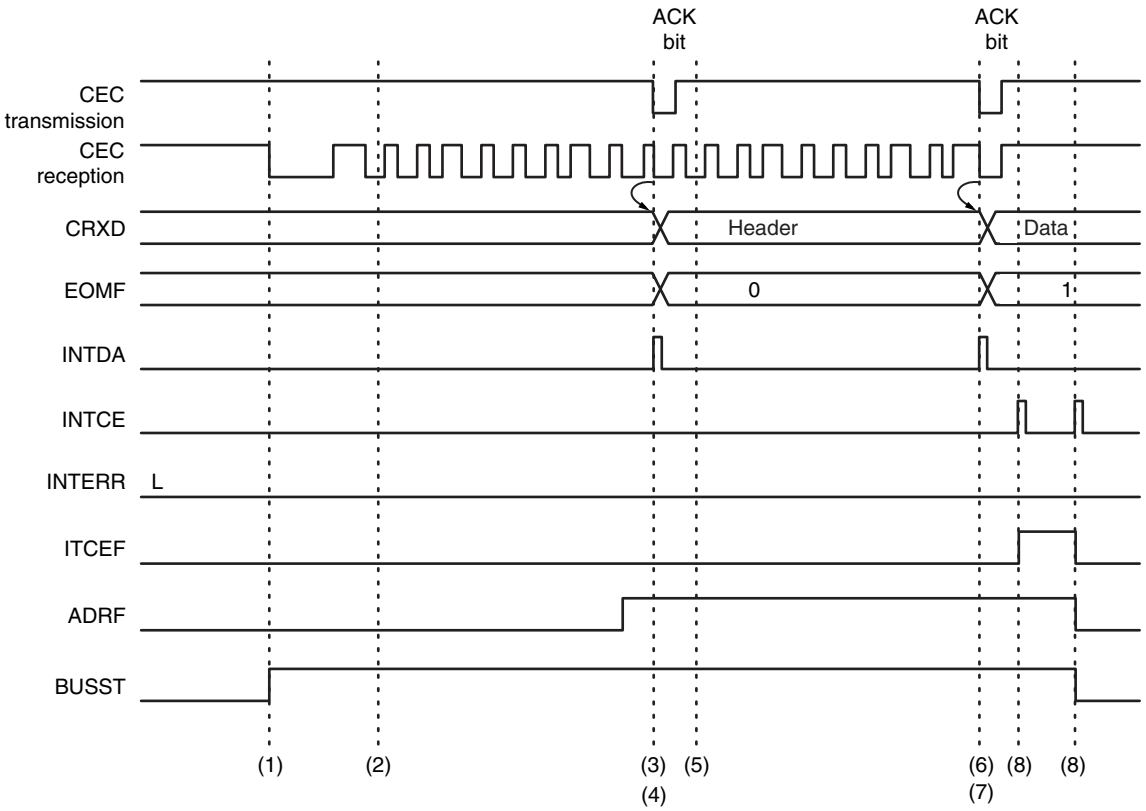
3. Only, INTCE output is generated if an error is detected.

4. This is supported only if detecting timing errors for the start bit (STERRD = 1). An error is detected but a flag is not set.

Remark √: Supported, ×: Not supported, –: Don't care

(1) CEC reception manipulation procedure

Figure 13-51. Basic Reception Timing (1)
(Direct Address Reception, CESEL0, CESEL1 = 0, 0)



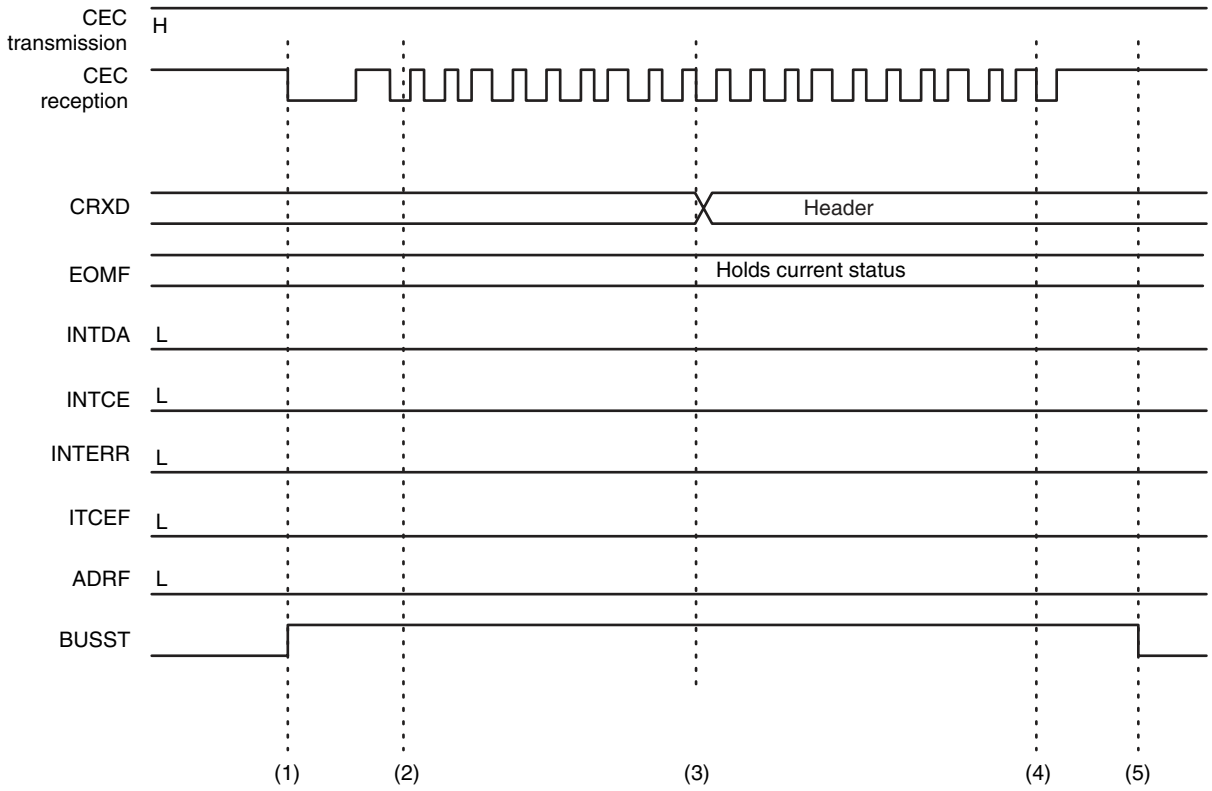
Caution A new start bit detected during transmission (the time from when a start bit is detected until the EOM of the last frame is received) is ignored and detected as a timing error.

Remark (1) to (8) in Figure 13-51 correspond to (1) to (8) in Figure 13-52.

Figure 13-52. CEC Reception Manipulation Procedure (1)

	Software Manipulation	Hardware State
CEC receive operation		[Start bit detection] The falling edge of the CEC reception signal is detected and a receive operation is started (1). [Sampling] The data is sampled at the NOMT setting time and sequentially stored in the shift register (2). [Address match interrupt] INTDA is generated because the address received at the header block has matched with the local address (3).
	Prepare for receiving a standby release or the like in response to INTDA generation. Use EOMF to check whether to continue transmission or whether it is the last frame.	[ACK bit transmission] Logical 0 is transmitted at the ACK bit timing because the reception was successful (4). [Continuing reception] The data of the second frame is continuously received (5). [Receive data interrupt] When receiving 8-bit data is completed, the data is transferred to CRXD and INTDA is generated (6).
	Read the receive data from CRXD in response to INTDA generation. Use EOMF to check whether to continue transmission or whether it is the last frame.	[ACK bit transmission] Logical 0 is transmitted at the ACK bit timing because the reception was successful (7). [Reception completion] The reception is judged to be completed because EOM = 1 is received and INTCE is output according to the CESEL1, CESEL0, SFT1, and SFT0 bit settings (8).

Figure 13-53. Basic Reception Timing (2)
(CECRXEN = 1, Direct Address, Address Mismatch, CINTMK = 0)



Remark (1) to (5) in Figure 13-53 correspond to (1) to (5) in Figure 13-54.

Figure 13-54. CEC Reception Manipulation Procedure (2)

	Software Manipulation	Hardware State
CEC receive operation		[Start bit detection] The falling edge of the CEC reception signal is detected and a receive operation is started. Set BUSST flag (1). [Sampling] The data is sampled at the NOMT setting time and sequentially stored in the shift register (2). [Address match interrupt] INTDA is not generated and neither ACK nor NACK is returned because the address received at the header block does not match the local address (3). However, monitoring CECRXD continues in order to check the bit length and detect communication completion. [ACK bit transmission] Neither ACK nor NACK is returned because communication is performed between other stations (4). [Reception completion] Communication between other stations is judged to be completed because EOM = 1 is received, the signal-free time is counted according to the SFT1 and SFT0 bit settings, and then BUSST is cleared to 0 (5).

(2) Broadcast reception

The reception flow and timing check period are the same as those of direct address reception. If the destination address transmitted by the initiator is FH, the communication operates as a broadcast reception.

The differences from direct address reception are as follows.

- Logical 1 is transmitted at the ACK bit timing in a normal operation.
- If reception has failed or CECRXEN = 0 has been set, logical 0 is transmitted at the ACK bit timing.

(3) CEC reception interrupt

Three interrupt functions, namely a data interrupt (INTDA), communication complete interrupt (INTCE), and error interrupt (INTERR) are provided.

A data interrupt (INTDA) is output at the following timings during reception (follower).

- When the address received at the header block of a direct address communication has matched the local address
- When address reception has been completed at the header block of a direct address communication when CINTMK = 1 is set
- When address reception of a broadcast communication has been completed at the header block
- When data reception has been completed at the data block and the receive data has been stored in the CRXD register

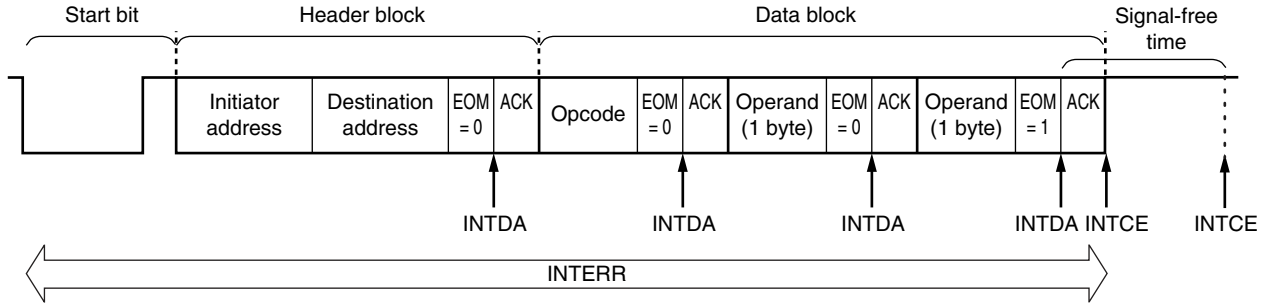
Communication complete interrupt INTCE is output at the following timings during reception (follower).

- CESEL1, CESEL0 = 0, 0
  INTCE is output if receiving the ACK bit of the last frame (EOM = 1) ends, if the signal-free time has been counted, or if the falling edge of the CEC line is detected starting in the high-level period of the ACK bit of the last frame while the signal-free time is counted.
- CESEL1, CESEL0 = 0, 1
  INTCE is output if receiving the ACK bit of the last frame (EOM = 1) ends or if the falling edge of the CEC line is detected starting in the high-level period of the ACK bit of the last frame while the signal-free time is counted.
- CESEL1, CESEL0 = 1, 0
  INTCE is output if the signal-free time has been counted.

An error interrupt (INTERR) is output at the following timings during reception (follower).

- When a timing error is detected
- When an overrun error is detected
- When a bus lock error is detected when BLERRD = 1 is set

Figure 13-55. Basic Reception Interrupt Timing



13.7.6 Error detection function

The hardware detects the following seven errors.

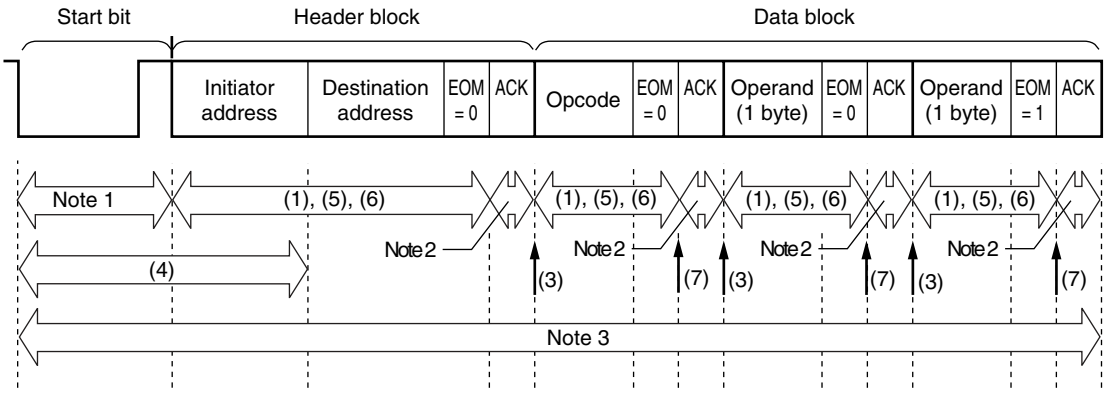
The errors that can be detected for the initiator and follower are shown below.

Table 13-4. Errors That Can Be Detected for Initiator and Follower

Error	Initiator	Follower
(1) Transmission error	√	×
(2) ACK error	√	×
(3) Underrun error	√	×
(4) Arbitration error	√	×
(5) Timing error (low-level width)	√	√
(6) Timing error (bit width)	√	√
(7) Overrun error	×	√
(8) Bus lock error	√	√

Remark √: Detected, ×: Not detected

Figure 13-56. Error Detection Period



Notes 1. (5) and (6) if detecting timing errors for the start bit (STERRD = 1).

2. (6) if detecting timing errors for the ACK bit (ACKTEN = 1), in addition to (2) and (5).

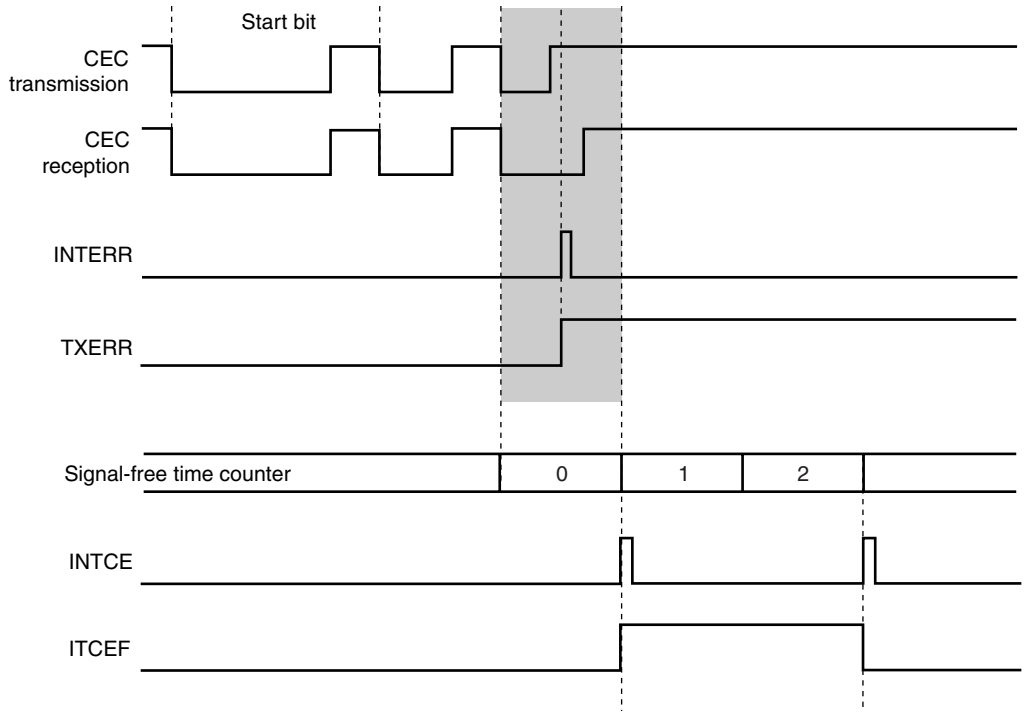
3. (8) if detecting bus lock errors (BLERRD = 1).

The details of each error are explained below.

(1) Transmission error

During initiator operation, the logic of the data the initiator has transmitted is compared with that of the CEC line receive data and a transmission error occurs when they vary. Errors are judged at the timing of the value set to the CEC reception data sampling time setting register (NOMT). Errors are judged during the data bit period of the frame that includes the EOM bit. After an error has been detected, an error interrupt (INTERR) is generated, the transmission error flag (TXERR) is set, and the transmission is stopped at that bit. INTCE is generated at the end of bit at which transmission stopped and after the signal-free time has been counted, according to the values specified for CESEL1 and CESEL0.

Figure 13-57. Transmission Error Detection Waveform (When Three Bits Are Set as Signal-Free Time)



When a transmission error is detected, the transmit operation is stopped at the bit where the error was detected, regardless of the set value of the EOM bit of the CECCTL0 register.

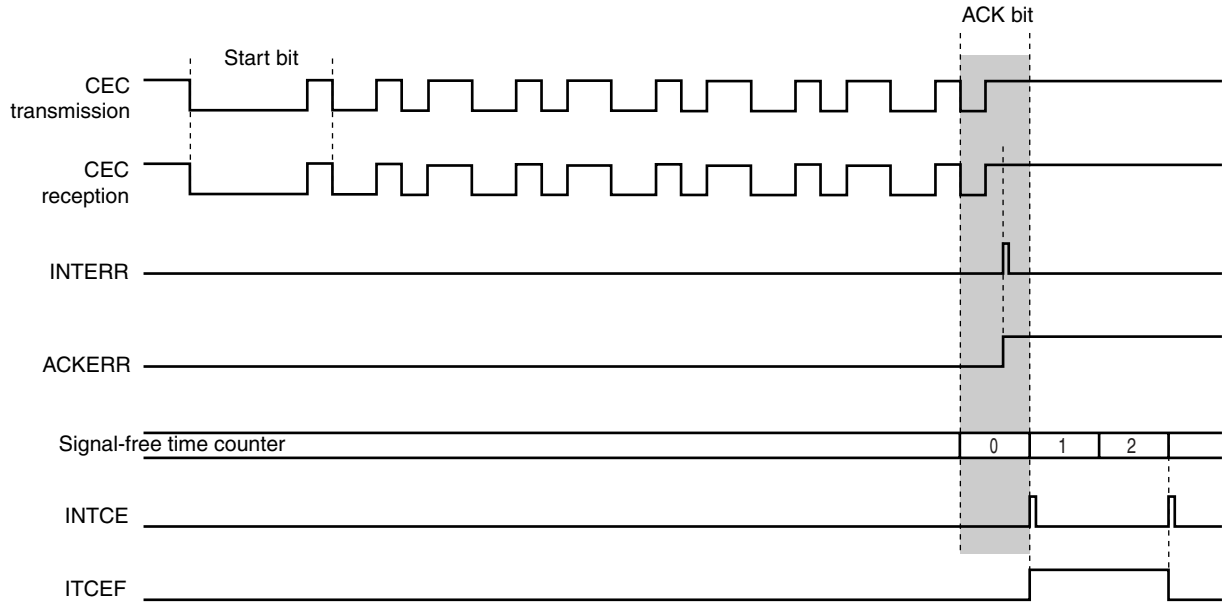
If EOM = 0 is received even though the initiator is transmitting EOM = 1, the transmission is judged as a transmission error and stopped. The follower judges that the transmission be continued, because EOM = 0. If BLERRD is set to 1, whether the received data is staying at high or low level can be detected.

(2) ACK error

During direct address transmission, an ACK error occurs when the initiator receives logical 1 at the ACK bit timing. During broadcast transmission, an ACK error occurs when the initiator receives logical 0 at the ACK bit timing.

Errors are judged at the timing of the value set to the CEC reception data sampling time setting register (NOMT). After an error is detected, an error interrupt (INTERR) is generated and the ACK error flag (ACKERR) is set. After the end of the ACK bit, the communication standby state is entered and the signal-free time is counted. INTCE occurs once or twice depending on the set values of CESEL1 and CESEL0.

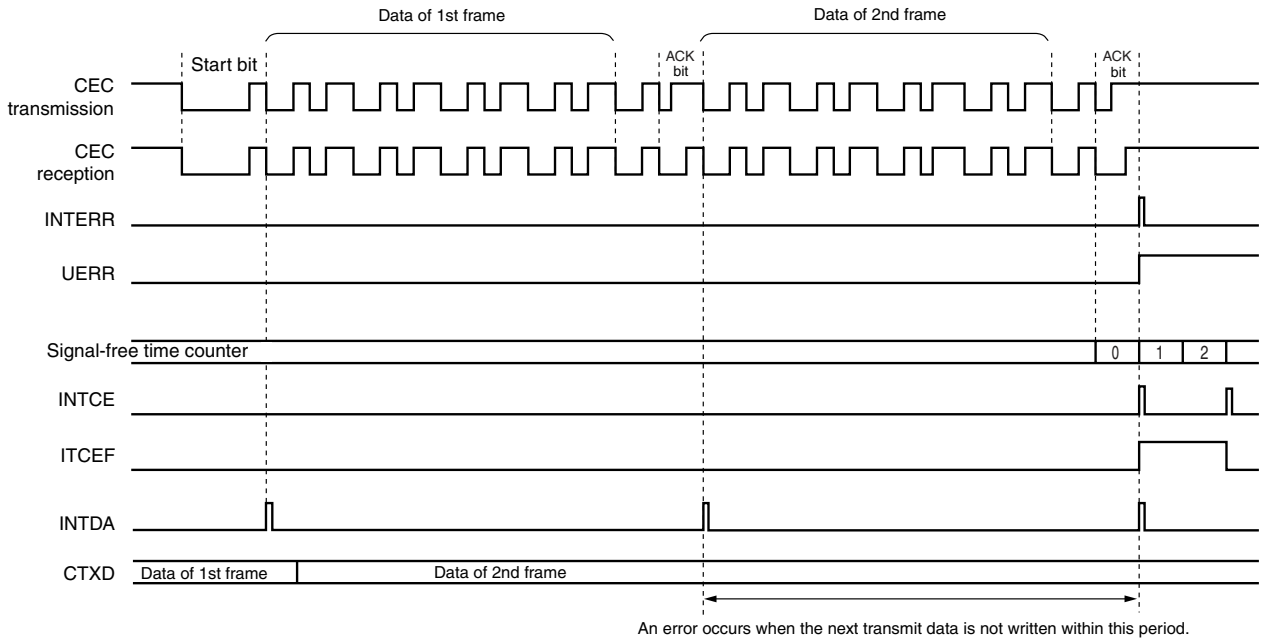
Figure 13-58. ACK Error During Direct Address Communication (When Three Bits Are Set as Signal-Free Time)



(3) Underrun error

An underrun error occurs when no data is set to the transmission buffer when transmitting the next data is started. An error interrupt (INTERR) is generated, the underrun error flag (UERR) is set, the transmission is aborted, and the communication standby state is entered. INTCE occurs once or twice depending on the set values of CESEL1 and CESEL0.

Figure 13-59. Underrun Error Timing



(4) Arbitration error

If logical 0 is received in response to logical 1 transmission during the period from the transmission start flag being set to the source address being transmitted, an arbitration error occurs. Errors between setting the transmission start trigger and outputting the start bit are judged when low level is output to the CEC transmission signal. While the source address is being transmitted, errors are judged at the timing of the value set to the CEC reception data sampling time setting register (NOMT). After an error is detected, an error interrupt (INTERR) is generated and the arbitration error flag (AERR) is set. At this time, the transmission is aborted, but the receive operation is continued. Multiple error flags may be detected, as shown in Figure 13-60, until the source address detection period is entered.

Figure 13-60. Arbitration Timing

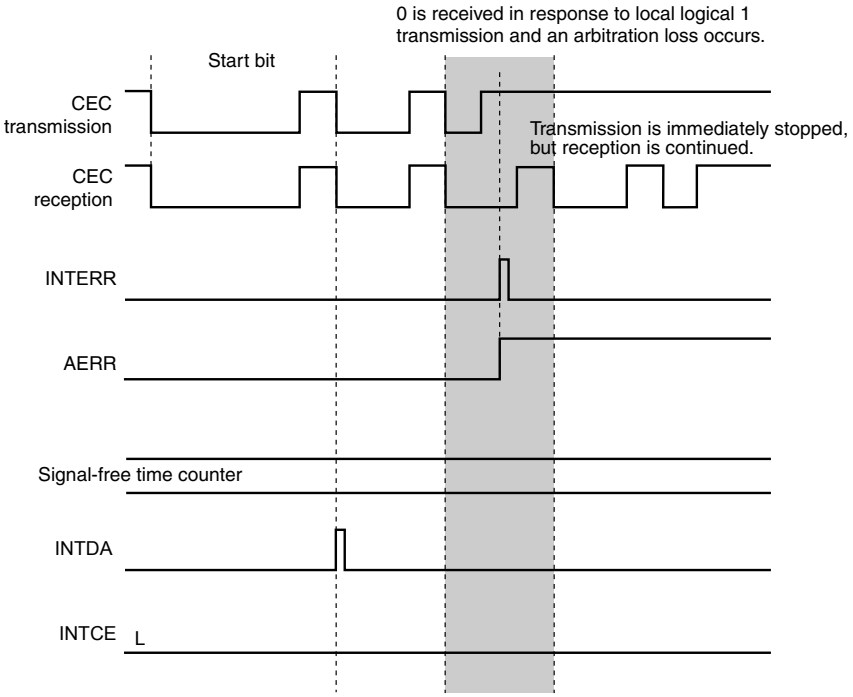
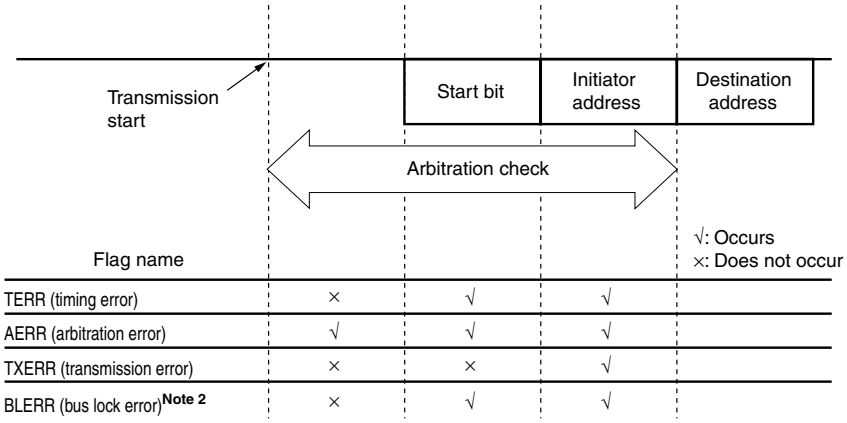


Figure 13-61. Relationships Between Arbitration Error and Other Errors



Notes 1. If STERRD is 1, the start bit timing is checked, but no error interrupt is generated.

2. If BLERRD is 1, a bus lock error is detected.

[Detailed explanation of arbitration error]

Details about the arbitration check performed from the time when the transmission start flag is set until the initiator address output period are provided below.

(a) Arbitration check by setting transmission start flag

The arbitration check is performed when two clocks have elapsed after the transmission start flag is set. If an arbitration loss is judged, an error interrupt (INTERR) is generated, the arbitration loss flag (AERR) is set, and the mode is immediately switched to reception mode.

(b) Start bit output period

If low level is detected at the reception line when the transmission start flag is set and the start bit is actually output, the arbitration loss flag is set and the mode is switched to reception mode. If STERRD = 1 and if the rising edge of the reception line is detected beyond the maximum low-level width of the start bit set by STATLH, the arbitration loss flag is set and the mode is switched to reception mode.

(c) Initiator address output period

After transmitting of the start bit is completed, a logic check is performed at the same time as starting to transmit the Initiator address. If an address earlier than the local address is detected, an error interrupt (INTERR) is generated, the timing error flag (TERR) and arbitration loss flag (AERR) are set, and the mode is immediately switched to reception mode.

(5) Timing error

Timing errors of the CEC reception signal are checked during whether initiator or follower operation. A timing error occurs if the CEC reception signal is outside the range of the compare register set. Low-level width timing errors are detected when the rising edge is detected. Timing errors for the minimum bit width are detected when the falling edge is detected, and timing errors for the maximum bit width are detected if there is no falling edge even though the maximum bit width has been exceeded. Whether to check the ACK bit timing can be selected using the ACKTEN bit of the CECCTL0 register. However, even if ACKTEN is set to 1, the maximum bit width is not checked only for the ACK bit of the last data block (when EOM is 1). The minimum bit width is checked. If a timing error with a short bit width is detected during follower operation, a low-level pulse (error handling pulse) that has a bit width 1.5 times the bit width specified using the NOMP register is transmitted. If a timing error with a short bit width is detected during initiator operation, transmission immediately stops and then the communication standby state is entered.

An error handling pulse is not transmitted if a start bit timing error is detected.

If a timing error other than one that has a short bit width is detected during initiator operation, transmission immediately stops. Reception is continued and a NACK is transmitted at the ACK bit timing during follower operation.

The generation of INTCE depends on the set values of CESEL1 and CESEL0.

If a timing error for the minimum bit width of the last ACK bit is detected, INTCE is output at the same time as INTERR.

Figure 13-62. Timing Error When Bit Width Is Short During Initiator Operation

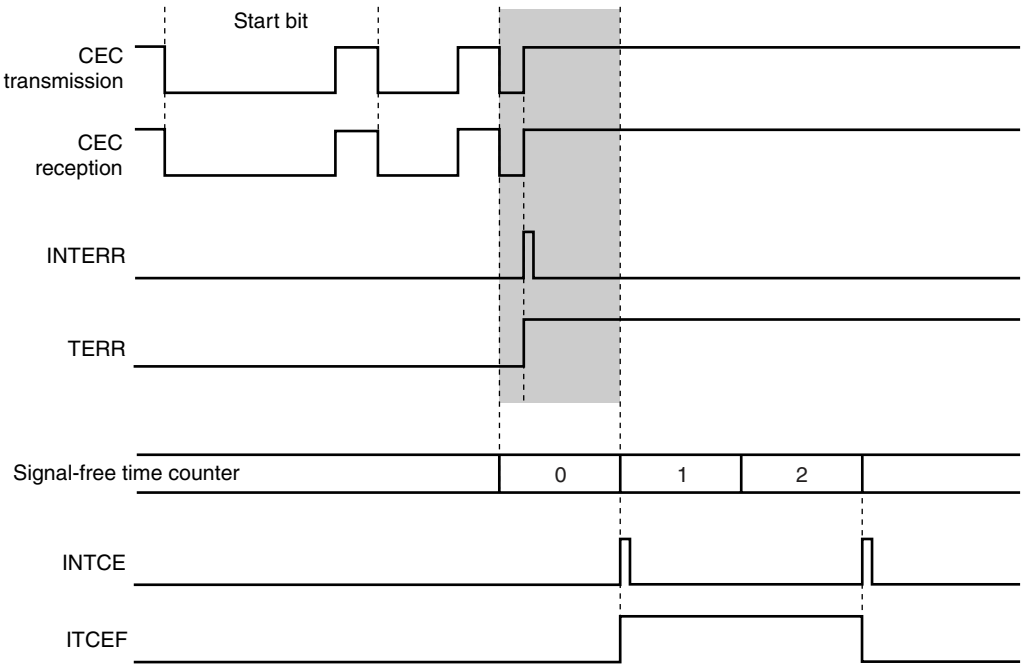


Figure 13-63. Timing Error When Bit Width Is Short During Follower Operation

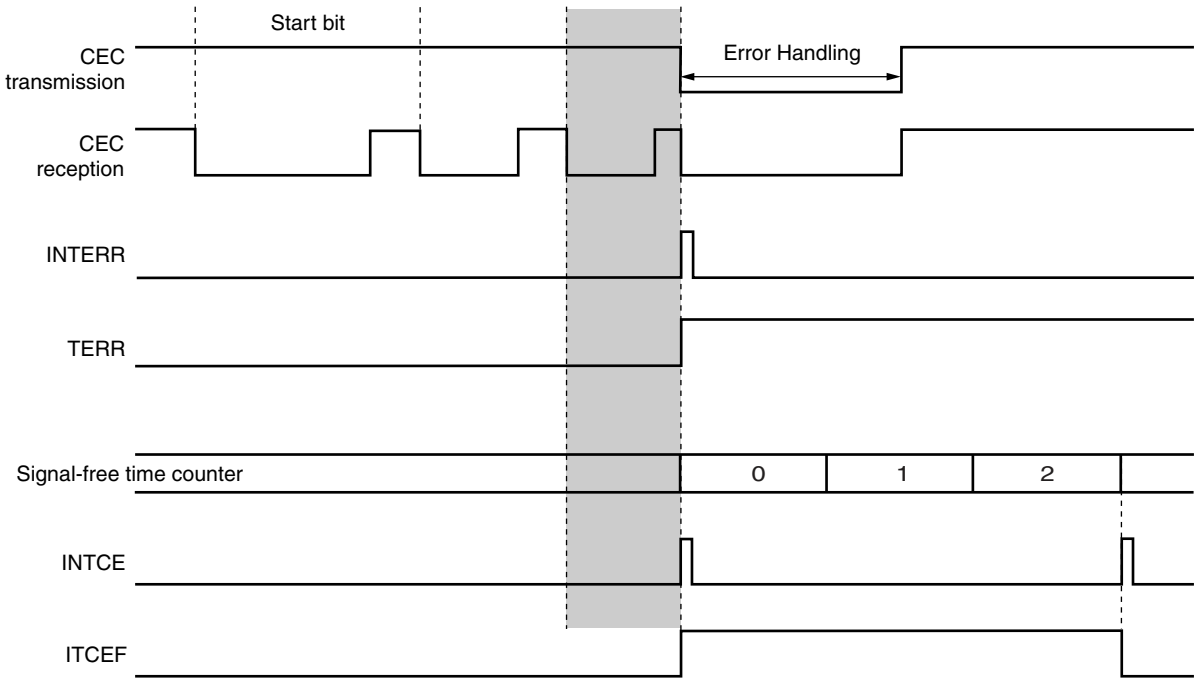
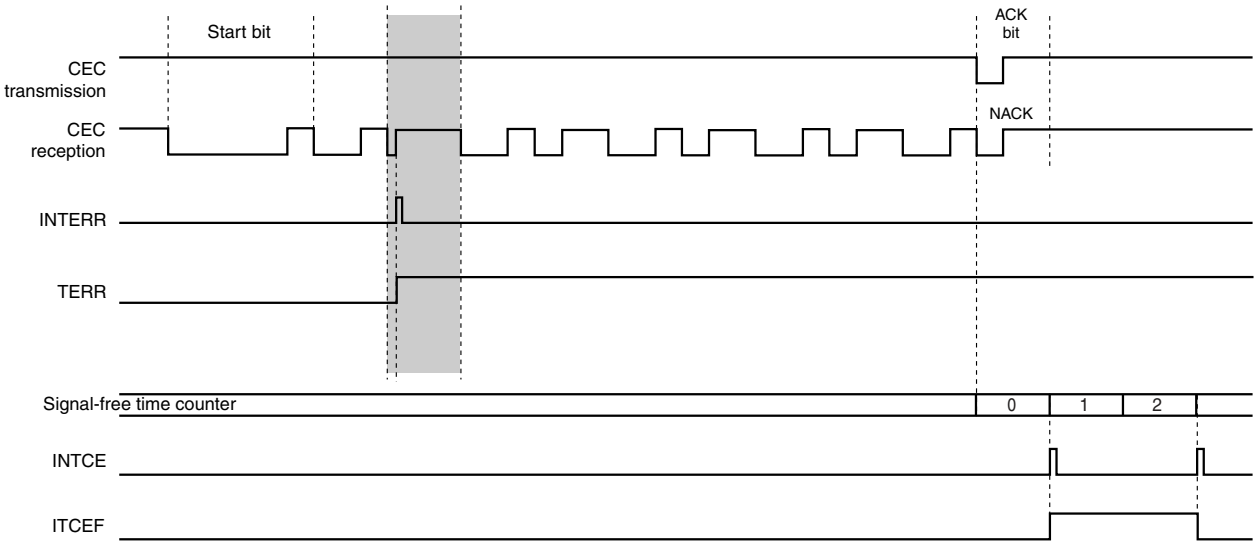


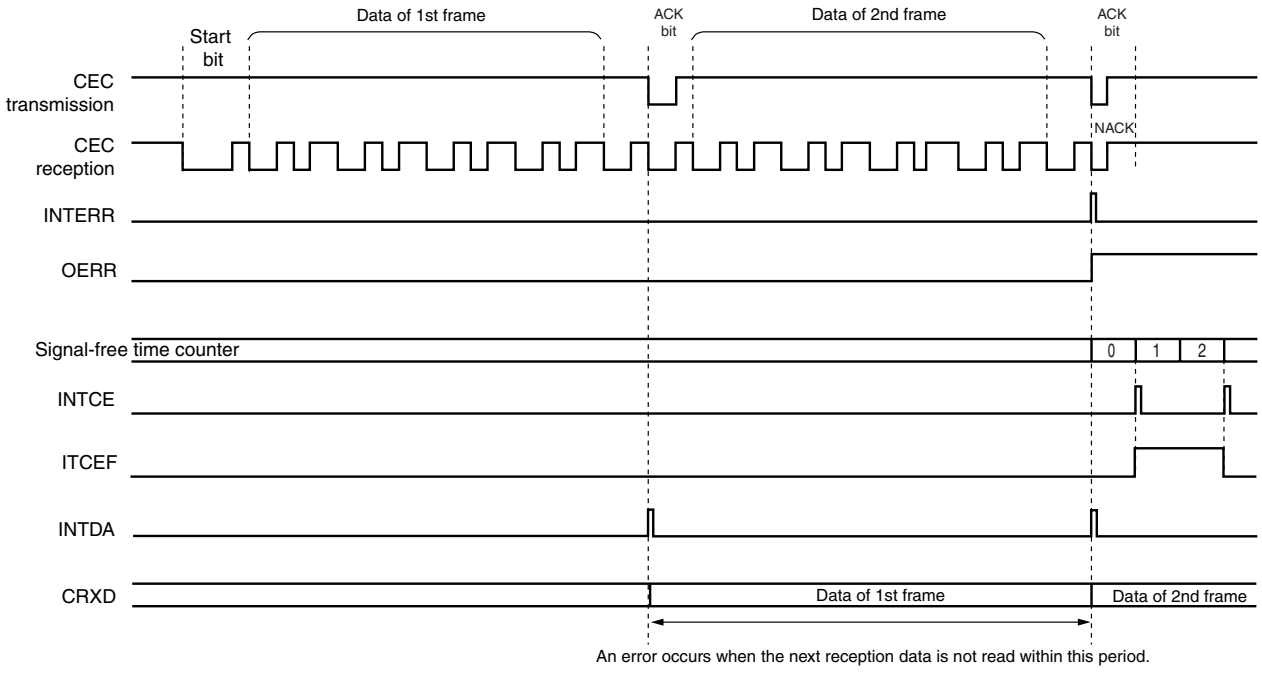
Figure 13-64. Timing Error When Bit Width Is Short During Follower Operation



(6) Overrun error

If receiving the next data is completed before reading data from the reception buffer register (CRXD) during follower operation, an overrun error occurs. An error interrupt (INTERR) is generated, the overrun error flag (OERR) is set, and the CRXD buffer value is overwritten by a new value. Afterward, logical 1 is returned during direct address communication and logical 0 is returned during broadcast communication at the ACK transmission timing of the block in which an overrun error occurred. The failure of reception is reported to the initiator, and the reception standby state is entered. INTCE operates according to the setting of CESEL1 and CESEL0.

Figure 13-65. Overrun Error (When Three Bits Are Set as Signal-Free Time)



(7) Bus lock error

A bus lock error occurs if the CEC reception signal stays at high or low level for a period corresponding to 2.5 times the data bit width specified using NOMP[Note]. Whether an input signal is stuck to high/low level can be detected by setting BLERRD to 1. If an error is detected, an error interrupt (INTERR) is generated, the bus lock error flag (BLERR) is set, the communication standby state is entered, and the signal-free time is counted. INTCE operates according to the setting of CESEL1 and CESEL0.

Note The above applies after the falling edge is detected and communication is started.

Caution If bus lock errors are not to be detected, the communication may not be terminated by an intended operation. In that case, the retransmission timing is lost and abnormal communication is continued. Consequently, set BLERRD and set, so that bus lock errors are detected during a receive operation.

Figure 13-66. During Initiator Operation (When Three Bits Are Set as Signal-Free Time)

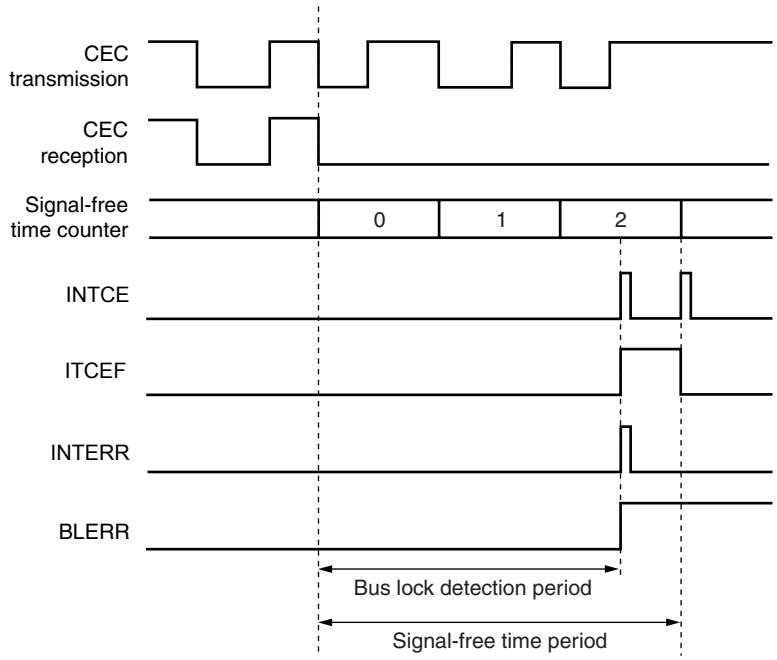
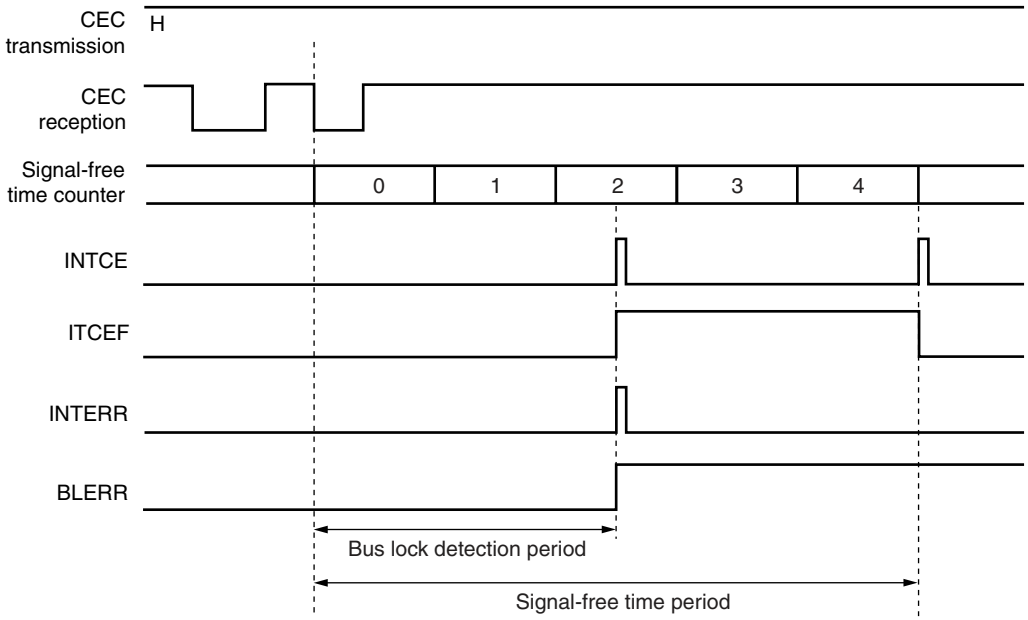


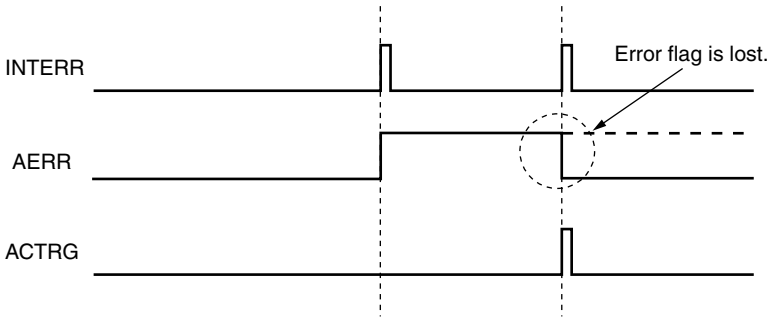
Figure 13-67. During Follower Operation (When Five Bits Are Set as Signal-Free Time)



13.7.7 Clearing error flag

An error flag set to the CEC communication error status register (CECES) can be cleared by setting 1 to the corresponding bit of the CEC communication error flag clear trigger register (CECFC). Figure 13-68 shows the case when an arbitration error occurs. An arbitration error flag can be cleared by setting 20H to CECFC. An error flag that has been set once will be cleared by using the clear trigger register or setting CECE = 0.

Figure 13-68. When Two Identical Errors Have Occurred and Conflict with Clear Trigger



Caution If clearing an error flag and setting an error flag to be cleared conflict, the second error flag may not be set.

13.7.8 Signal-free time

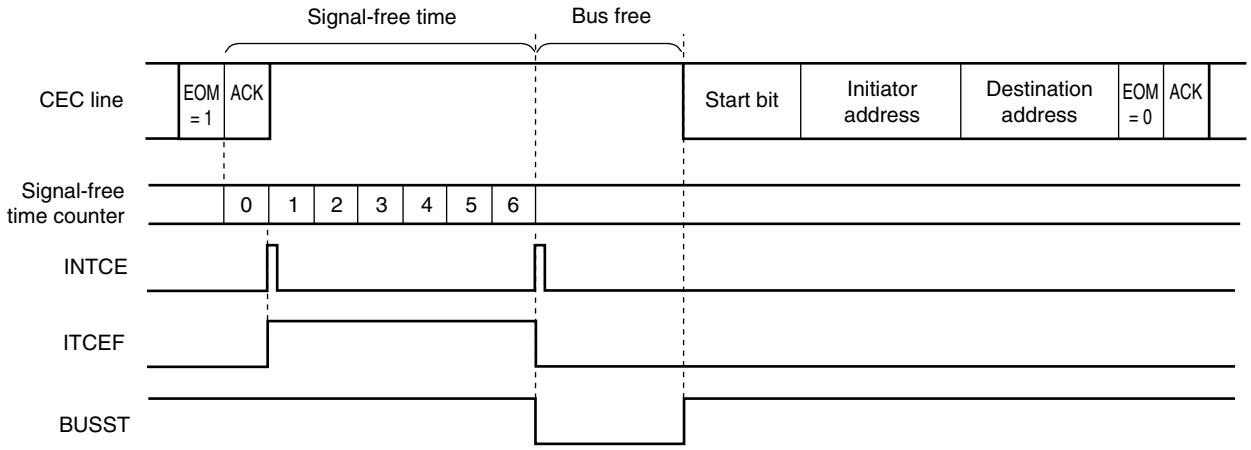
The end of the signal-free time is reported by generating a communication complete interrupt by detecting a match with the specified time (3, 5, or 7 bits of the bit width specified using the NOMP register). The number of bits of the signal-free time is specified using SFT1 and SFT0. When a communication complete interrupt is generated can be selected by setting up CESEL1 and CESEL0. Counting the signal-free time always starts when the falling edge of the received data is detected. During normal communication, counting the signal-free time starts after the falling edge of the ACK bit is detected, if EOM is 1.

If the communication standby state is forcibly entered due to the occurrence of an error, counting is started as soon as the communication standby state is entered.

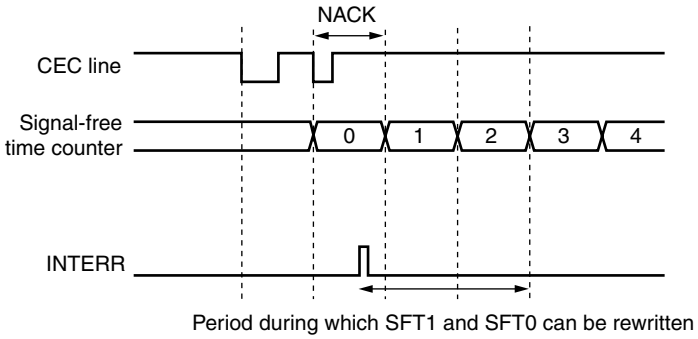
If an error handling pulse (a low-level pulse that has a bit width 1.5 times the bit width specified using the DATB register) is acknowledged, the signal-free time is counted starting at the falling edge of the error handling pulse signal.

Figure 13-69 shows an operation example in which the detection of a 7-data bit width signal-free time that is set to CESEL1 = 1, CESEL0 = 0, SFT1 = 1, and SFT0 = 0 is set.

Figure 13-69. Signal-Free Time Operation



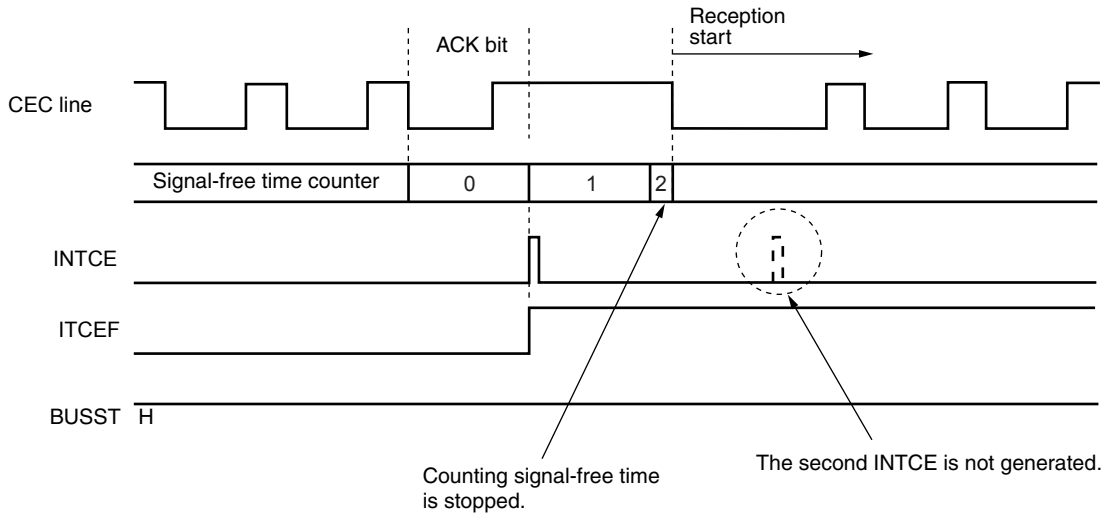
Caution Rewriting the values of the SFT1 and SFT0 registers to the number of bits smaller than the current number of bits while counting the signal-free time must be completed until the rewritten bit count values match. If rewriting is not performed in time, the counter overflows and the signal-free time period continues until the number of bits match again. An example in which the data bit width is changed from five to three bits is shown below.



<Starting receive operation during signal-free time>

If a falling edge of the CEC reception signal is detected while counting the signal-free time, a receive operation is started. INTCE is not output even if the output of INTCE after counting the signal-free time is set, because the count operation of the signal-free time counter is stopped at this time.

Similarly, if 1 is written to the transmission trigger while counting the signal-free time, a transmission is started and the count operation of the signal-free time counter is stopped.



CHAPTER 14 REMOTE CONTROLLER RECEIVER

14.1 Remote Controller Receiver Functions

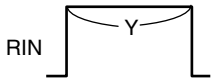
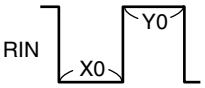
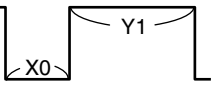
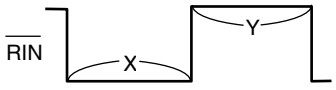
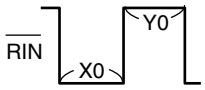
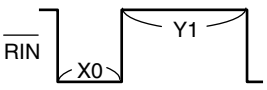
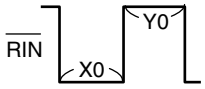
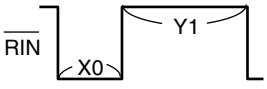
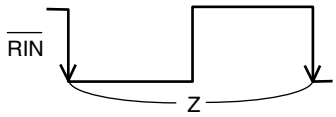
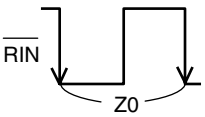
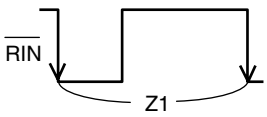
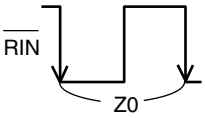
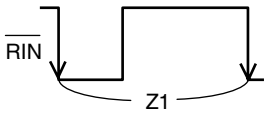
- Multiple channels supported (4 channels are provided on chip)
- Bit detection by high-/low-level width or cycle of remote control reception waveform
- Continuous reception even during reception error occurrence
- Noise elimination function (pulse noise elimination/cycle noise elimination)

The remote controller receiver uses the following five remote controller reception modes.

- Type A reception mode … Bit detection by high-/low-level width, guide pulse (half clock) provided
- Type B reception mode … Bit detection by high-/low-level width, guide pulse (1 clock) provided
- Type C reception mode … Bit detection by high-/low-level width, guide pulse not provided
- Type B1 reception mode … Bit detection by cycle, guide pulse (1 cycle) provided
- Type C1 reception mode … Bit detection by cycle, guide pulse not provided

Remark Guide pulse (reader code): Remote control signal header section

Table 14-1. Format of Each Reception Mode (Waveform to Be Input to RMIN)

Reception mode	Guide pulse	Data 0	Data 1	Example of Format
Type A				See **14. 5. 1**
Type B				See **14. 5. 3**
Type C	Not provided			See **14. 5. 5**
Type B1				See **14. 5. 7**
Type C1	Not provided			See **14. 5. 9**

Remarks 1. The waveform of data 1 is longer than that of data 0 (Y0 < Y1, Z0 < Z1).

2. X, X0: low-level width
Y, Y0, Y1: high-level width
Z, Z0, Z1: cycle

3. RIN: Input signal from the remote control receive data input (RIN01, RIN23) pin
$\overline{\text{RIN}}$: Inverted RIN signal
RMIN: See **Figure 14-1**.

14.2 Remote Controller Receiver Configuration

The remote controller receiver includes the following hardware.

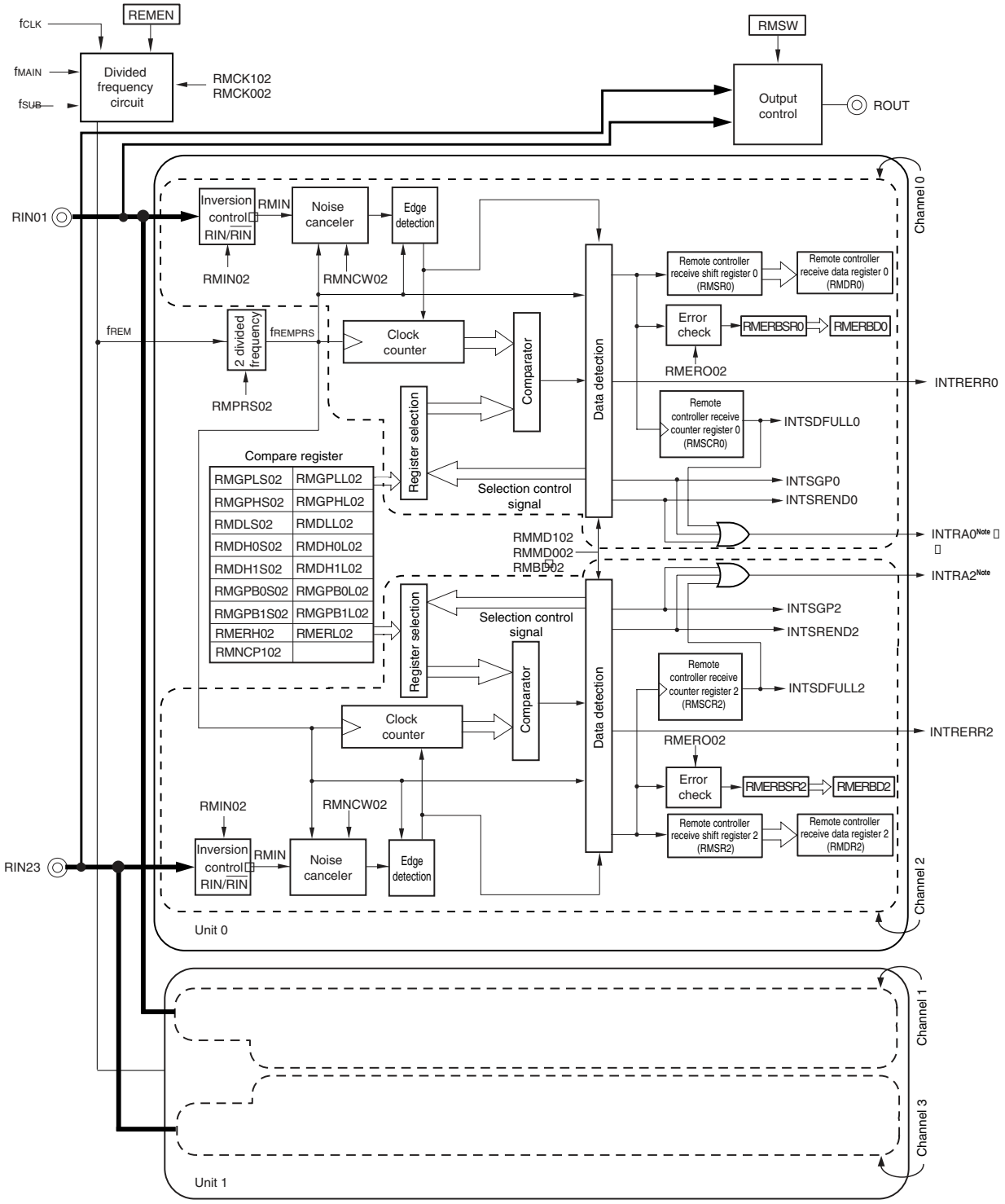
Table 14-2. Remote Controller Receiver Configuration (Unit 0)

Item	Configuration
Registers dedicated to channel 0	Remote controller receive data register 0 (RMDR0) Remote controller receive counter register 0 (RMSCR0) Remote controller receive shift register 0 (RMSR0) Remote controller receive interrupt status register 0 (RMINTS0) Remote control reception error bit detection register 0 (RMERBD0) Remote control reception error bit detection shift register 0 (RMERBSR0)
Registers dedicated to channel 2	Remote controller receive data register 2 (RMDR2) Remote controller receive counter register 2 (RMSCR2) Remote controller receive shift register 2 (RMSR2) Remote controller receive interrupt status register 2 (RMINTS2) Remote control reception error bit detection register 2 (RMERBD2) Remote control reception error bit detection shift register 2 (RMERBSR2)
Common registers	Peripheral enable register 1 (PER1) Remote controller receive control 1 register 02 (RMCN102) Remote controller receive control 2 register 02 (RMCN202) Remote controller receive data slew control register (RMSW) Remote controller receive GPLS compare register 02 (RMGPLS02) Remote controller receive GPLL compare register 02 (RMGPLL02) Remote controller receive GPHS compare register 02 (RMGPHS02) Remote controller receive GPHL compare register 02 (RMGPHL02) Remote controller receive DLS compare register 02 (RMDLS02) Remote controller receive DLL compare register 02 (RMDLL02) Remote controller receive DH0S compare register 02 (RMDH0S02) Remote controller receive DH0L compare register 02 (RMDH0L02) Remote controller receive DH1S compare register 02 (RMDH1S02) Remote controller receive DH1L compare register 02 (RMDH1L02) Remote controller receive GPBS compare register 02 (RMGPBS02) Remote controller receive GPBL compare register 02 (RMGPBL02) Remote controller receive DB0S compare register 02 (RMDB0S02) Remote controller receive DB0L compare register 02 (RMDB0L02) Remote controller receive DB1S compare register 02 (RMDB1S02) Remote controller receive DB1L compare register 02 (RMDB1L02) Remote controller receive end width select register 02 (RMERH02 and RMERL02) Remote controller receive noise elimination period setting register 02 (RMNCP102) Port mode registers 4, 9 (PM4, PM9) Port registers 4, 9 (P4, P9)

Table 14-3. Remote Controller Receiver Configuration (Unit 1)

Item	Configuration
Registers dedicated to channel 1	Remote controller receive data register 1 (RMDR1) Remote controller receive counter register 1 (RMSCR1) Remote controller receive shift register 1 (RMSR1) Remote controller receive interrupt status register 1 (RMINTS1) Remote control reception error bit detection register 1 (RMERBD1) Remote control reception error bit detection shift register 1 (RMERBSR1)
Registers dedicated to channel 3	Remote controller receive data register 3 (RMDR3) Remote controller receive counter register 3 (RMSCR3) Remote controller receive shift register 3 (RMSR3) Remote controller receive interrupt status register 3 (RMINTS3) Remote control reception error bit detection register 3 (RMERBD3) Remote control reception error bit detection shift register 3 (RMERBSR3)
Common registers	Peripheral enable register 1 (PER1) Remote controller receive control 1 register 13 (RMCN113) Remote controller receive control 2 register 13 (RMCN213) Remote controller receive data slew control register (RMSW) Remote controller receive GPLS compare register 13 (RMGPLS13) Remote controller receive GPLL compare register 13 (RMGPLL13) Remote controller receive GPHS compare register 13 (RMGPHS13) Remote controller receive GPHL compare register 13 (RMGPHL13) Remote controller receive DLS compare register 13 (RMDLS13) Remote controller receive DLL compare register 13 (RMDLL13) Remote controller receive DH0S compare register 13 (RMDH0S13) Remote controller receive DH0L compare register 13 (RMDH0L13) Remote controller receive DH1S compare register 13 (RMDH1S13) Remote controller receive DH1L compare register 13 (RMDH1L13) Remote controller receive GPBS compare register 13 (RMGPBS13) Remote controller receive GPBL compare register 13 (RMGPBL13) Remote controller receive DB0S compare register 13 (RMDB0S13) Remote controller receive DB0L compare register 13 (RMDB0L13) Remote controller receive DB1S compare register 13 (RMDB1S13) Remote controller receive DB1L compare register 13 (RMDB1L13) Remote controller receive end width select register 13 (RMERH13 and RMERL13) Remote controller receive noise elimination period setting register 13 (RMNCP113) Port mode registers 4, 9 (PM4, PM9) Port registers 4, 9 (P4, P9)

Figure 14-1. Block Diagram of Remote Controller Receiver



Note The following three interrupt sources are output as INTRAn signals in this remote controller receiver.

- 8-bit data reception completion (INTDFULLn)
- Reception completion (INTRENDn)
- Guide pulse detection (INTGPn)

In this chapter, the interrupt sources are expressed as INTDFULLn, INTRENDn, and INTGPn for the sake of explanation, but should be read as follows.

- INTDFULLn: INTRAn interrupt occurrence and INTSDFULLn of the RMINTSn register = 1
- INTRENDn: INTRAn interrupt occurrence and INTSRENDn of the RMINTSn register = 1
- INTGPn: INTRAn interrupt occurrence and INTSGPn of the RMINTSn register = 1

Remark n = 0 to 3

(1) Remote controller receive shift register n (RMSRn)

This is an 8-bit register for reception of remote controller data.

Data is stored in bit 7 first. Each time new data is stored, the stored data is shifted to the lower bits. Therefore, the latest data is stored in bit 7, and the first data is stored in bit 0.

When data 0 is normally received, “0” will be stored in bit 7, and when data 1 is normally received, “1” will be stored in bit 7.

RMSRn is read with an 8-bit memory manipulation instruction.

Reset signal generation sets RMSRn to 00H.

Also, RMSRn is cleared to 00H under any of the following conditions.

- Remote controller stops operation (RMENm = 0).
- INTDFULLn is generated.
- RMSRn is read after INTRENDn has been generated.
- When an error occurs with type A, B, or C
- When RMEROm = 0 and an error occurs with type B1 or C1

Remark m = 02, 13, n = 0 to 3

Figure 14-2. Format of Remote Controller Receive Shift Register n (RMSRn)

Address: FFF42H, FFF54H, FFF5AH, FFF60H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMSRn								

Cautions 1. Reading RMSRn while receiving data is prohibited. Read RMSRn after INTRENDn is generated and the following registers are read.

- **RMSCRn**
- **RMERBSRn (type B1 or C1 reception mode only)**
- **RMERBDn (type B1 or C1 reception mode only)**

A value that has been read once cannot be guaranteed, because the contents of RMSRn will be cleared after the RMSRn read operation ends.

2. If eight bits of remote control signals are received by RMSRn, the contents of RMSRn will be transferred to remote control receive data register n (RMDRn) and INTDFULLn will be generated. At this time, RMSRn will be cleared.

3. After INTRENDn is generated, be sure to read RMSCRn before reading RMSRn.
RMSCRn and RMSRn will be automatically cleared after RMSRn is read.
After INTRENDn is generated, the next data cannot be received until RMSRn is read.

4. If operation continuation (RMEROm of the RMCN2m register = 1) is selected with type B1 or C1 and if an error occurs, "0" will be stored.
In this case, after INTRENDn is generated, be sure to read RMSCRn and RMERBSRn before reading RMSRn. RMSCRn, RMERBSRn, and RMSRn will be automatically cleared after RMSRn is read. After INTRENDn is generated, the next data cannot be received until RMSRn is read.

Remarks 1. m = 02, 13, n = 0 to 3

2. Even if the remote control signal format is other than an integral multiple of eight bits, remote control reception can be performed. The number of higher valid bits of the RMSRn register can be identified according to the value of remote control receive counter register n (RMSCRn). The bits other than the valid bits of the RMSRn register are 0.

(2) Remote controller receive data register n (RMDRn)

This register holds the remote controller reception data. When the remote controller receive shift register n (RMSRn) overflows, the data in RMSRn is transferred to RMDRn. Bit 7 stores the last data, and bit 0 stores the first data. INTDFULLn is generated at the same time as data is transferred from RMSRn to RMDRn.

RMDRn is read with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDRn to 00H.

When the remote controller operation is disabled (RMENm = 0), RMDRn is cleared to 00H.

Figure 14-3. Format of Remote Controller Receive Data Register n (RMDRn)

Address: FFF40H, FFF4EH, FFF58H,FFF5EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMDRn								

Caution When INTDFULLn has been generated, read RMDRn before the next 8-bit data is received. If the next INTDFULLn is generated before the read operation is complete, RMDRn is overwritten.

Remark m = 02, 13, n = 0 to 3

(3) Remote controller receive counter register n (RMSCRn)

This is a 3-bit counter register used to indicate the number of valid bits remaining in the remote controller receive shift register n (RMSRn) when remote controller reception is complete (INTRENDn is generated). The number of higher valid bits of the RMSRn register can be identified by reading the value of this register, even if remote control signals other than in a format, which is an integral multiple of eight bits, are received.

RMSCRn is read with an 8-bit memory manipulation instruction.

Reset signal generation sets RMSCRn to 00H.

It is cleared to 00H under any of the following conditions.

- Remote controller stops operation (RMENm = 0).
- RMSRn is read after INTRENDn has been generated.
- When an error occurs with type A, B, or C
- When RMEROm = 0 and an error occurs with type B1 or C1

Figure 14-4. Format of Remote Controller Receive Counter Register n (RMSCRn)

Address: FFF41H, FFF4FH, FFF59H, FFF5FH After reset: 00H R/W

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

Caution When INTRENDn has been generated, immediately read RMSCRn before reading RMSRn. If reading occurs at another timing, the value is not guaranteed.

Example of relationship between receive bit number and RMSCRn register

Receive bit number	RMSCRn register value	Receive bit number	RMSCRn register value
3	3 (3 – 8 × 0)	8	0 (8 – 8 × 1)
15	7 (15 – 8 × 1)	16	0 (16 – 8 × 2)

Remark m = 02, 13, n = 0 to 3

Table 14-4. Operation Examples of RMSRn, RMSCRn, and RMDRn Registers When Receiving 1010101011111111B (16 Bits)

	RMSRn								RMSCRn	RMDRn
	7	6	5	4	3	2	1	0		
After reset	0	0	0	0	0	0	0	0	00H	00000000B
Receiving 1 bit	1	0	0	0	0	0	0	0	01H	00000000B
Receiving 2 bits	0	1	0	0	0	0	0	0	02H	00000000B
Receiving 3 bits	1	0	1	0	0	0	0	0	03H	00000000B
…	…	…	…	…	…	…	…	…	…	…
Receiving 7 bits	1	0	1	0	1	0	1	0	07H	00000000B
Receiving 8 bits	0	1	0	1	0	1	0	1	00H	00000000B
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
RMDRn transfer	0	0	0	0	0	0	0	0	00H	01010101B
Receiving 9 bits	1	0	0	0	0	0	0	0	01H	01010101B
Receiving 10 bits	1	1	0	0	0	0	0	0	02H	01010101B
…	…	…	…	…	…	…	…	…	…	…
Receiving 16 bits	1	1	1	1	1	1	1	1	00H	01010101B
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
RMDRn transfer	0	0	0	0	0	0	0	0	00H	11111111B

Remark n = 0 to 3

(4) Remote control reception error bit detection shift register n (RMERBSRn) (type B1 or C1 reception mode only)

This is an 8-bit register storing the error bits that occurred upon remote control reception.

Data is stored in bit 7 first. Each time new data is stored, the stored data is shifted to the lower bits. Therefore, the latest data is stored in bit 7, and the first data is stored in bit 0.

When an error occurs, “1” will be stored in bit 7, and when data is normally received, “0” will be stored in bit 7.

This register can be read only by an 8-bit manipulation instruction.

Reset signal generation sets RMERBSRn to 00H.

It is cleared to 00H under any of the following conditions.

- Remote controller stops operation (RMENm = 0).
- RMSRn is read.

Figure 14-5. Format of Remote Control Reception Error Bit Detection Shift Register n (RMERBSRn)

Address: FFF4CH, FFF56H, FFF5CH, FFF62H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMERBSRn								

Cautions 1. After INTRENDn is generated, read this register before RMSRn. Reading this register during data reception is prohibited.

2. This register is valid when type B1 or C1 is used and 1 is specified for RMEROm of the RMCN2m register (operation continuation during error occurrence).

Remark m = 02, 13, n = 0 to 3

(5) Remote control reception error bit detection register n (RMERBDn) (type B1 or C1 reception mode only)

This register stores the error occurrence information with types B1 and C1.

If the remote control receive shift register overflows, the data of the remote control reception error bit detection shift register (RMERBSRn) will be transferred to the remote control reception error bit detection register.

This register can be read only by an 8-bit manipulation instruction.

Reset signal generation sets RMERBDn to 00H.

It is cleared to 00H under any of the following conditions.

- Remote controller stops operation (RMENm = 0).
- RMSRn is read after INTRENDn has been generated.

Figure 14-6. Format of Remote Control Reception Error Bit Detection Register n (RMERBDn)

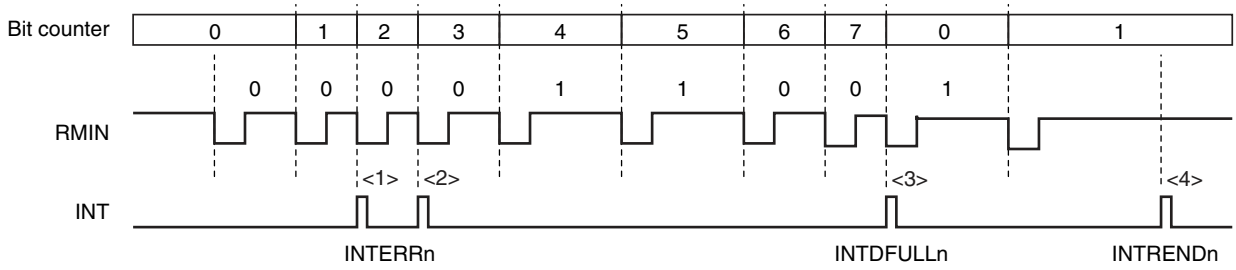
Address: FFF43H, FFF55H, FFF5BH, FFF61H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMERBDn								

Cautions 1. Read this register after an interrupt is generated, because it will be transferred after INTDFULLn is generated.

2. This register is valid when type B1 or C1 is used and 1 is specified for RMEROm of the RMCN2m register (operation continuation during error occurrence).

Example: If errors occur in bits 1, 2, and 7 (type B1 or C1, RMEROm = 1)



	Bit counter	RMERBSRn register	RMERBDn register	RMDRn register
<1>	2	80H	00H	00H
<2>	3	C0H	00H	00H
<3>	0	00H (transfer and clear)	86H	30H
<4>	1	00H (transfer and clear)	00H	80H

Remark m = 02, 13, n = 0 to 3

14.3 Registers to Control Remote Controller Receiver

The remote controller receiver is controlled by the following register.

- Peripheral enable register 1 (PER1)
- Remote controller receive control 1 register m (RMCN1m)
- Remote controller receive control 2 register m (RMCN2m)
- Remote controller receive interrupt status registers n (RMINTSn)
- Remote controller receive data slew control register (RMSW)
- Port mode registers 4, 9 (PM4, PM9)
- Port registers 4, 9 (P4, P9)

Remark m = 02, 13, n = 0 to 3

(1) Peripheral enable registers 1 (PER1)

This register is 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.

When the remote controller receiver is used, be sure to set bit 5 (REMEN) of this register to 1.

PER1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears PER1 to 00H.

Figure 14-7. 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	REMEN	CECEN	0	0	0	0

REMEN	Control of remote controller receiver input clock supply
0	Stops input clock supply. • SFR used by remote controller receiver cannot be written. • Remote controller receiver is in the reset status.
1	Supplies input clock. • SFR used by remote controller receiver can be read and written.

Cautions 1. When using the remote controller receiver, first set REMEN to 1. If REMEN = 0, writing to a control register of the remote controller receiver is ignored, and, even if the register is read, only the default value is read.

2. Be sure to clear bits 0 to 3, 6, and 7 of the PER1 register to 0.

(2) Remote controller receive control 1 register m (RMCN1m)

This register is used to select the remote controller receive operation mode.

RMCN1m is set with a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets RMCN1m to 00H.

Figure 14-8. Format of Remote Controller Receive Control 1 Register m (RMCN1m) (1/2)

Address: F0330H, F0342H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMCN102	RMEN02	RMNCW02	RMPRS02	RMIN02	RMMD102	RMMD002	RMCK102 Note 1	RMCK002 Note 1
RMCN113	RMEN13	RMNCW13	RMPRS13	RMIN13	RMMD113	RMMD013	0	0

RMENm	Enable remote controller reception
0	Disable remote controller reception (default)
1	Enable remote controller reception

RMNCWm Note 2	Noise elimination width control (eliminate pulse)
0	Eliminate noise less than 1/f_{REMPRS} (default)
1	Eliminate noise less than 2/f_{REMPRS}

RMPRSm Note 2	Clock division control within each channel
0	Clock not divided within each channel (f_{REMPRS} = f_{REM}) (default)
1	Clock divided into two within each channel (f_{REMPRS} = f_{REM}/2)

Division control (RMPRSm Note 2)	Noise elimination width control (RMNCWm Note 2)	Internal operation clock frequency (1/f_{REMPRS}) Note 3	Noise width that can be eliminated
0	0	1/f_{REM}	Less than 1/f_{REM}
0	1	1/f_{REM}	Less than 2/f_{REM}
1	0	2/f_{REM}	Less than 2/f_{REM}
1	1	2/f_{REM}	Less than 4/f_{REM}

Caution Receiving remote control waveforms will be started if 1 is set to RMENm and five clocks of f_{REM} elapse. Remote control waveforms cannot be received during the period of the five f_{REM} clocks after 1 is set to RMENm.

Notes 1. The RMCK102 and RMCK002 bits are only provided in the RMCN102 register. Bits 1 and 0 of the RMCN113 register are read-only and 0 is read.

2. Set this bit when RMENm = 0.

3. f_{REMPRS}: Internal operation clock after division control by RMPRSm
f_{REM}: Clock selected by using RMCK102 and RMCK002

Remark m = 02, 13

Figure 14-8. Format of Remote Controller Receive Control 1 Register m (RMCN1m) (2/2)

Address: F0330H, F0342H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMCN102	RMEN02	RMNCW02	RMPRS02	RMIN02	RMMD102	RMMD002	RMCK102 Note 1	RMCK002 Note 1
RMCN113	RMEN13	RMNCW13	RMPRS13	RMIN13	RMMD113	RMMD013	0	0

RMINm Note 2	Remote controller input invert control
0	Does not invert signal of remote control receive data input pin (default).
1	Inverts signal of remote control receive data input pin.

RMBDm Notes 2, 3	RMMD1m Note 2	RMMD0m Note 2	Remote controller reception mode
0	0	0	Type A reception mode (guide pulse (half clock) provided) (default)
0	0	1	Type B reception mode (guide pulse (1 clock) provided)
0	1	0	Type C reception mode (guide pulse not provided)
0	1	1	Setting prohibited
1	0	0	Setting prohibited
1	0	1	Type B1 reception mode (guide pulse (1 cycle) provided)
1	1	0	Type C1 reception mode (guide pulse not provided)
1	1	1	Setting prohibited

RMCK102 Note 4	RMCK002 Note 4	Selection of source clock (f_{REM}) of remote controller counter				
		Selection clock	f_{MAIN} = 2 MHz	f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz
0	0	$f_{MAIN}/2^7$	15.625 kHz	39.063 kHz	78.125 kHz	156.250 kHz
0	1	$f_{MAIN}/2^8$	7.813 kHz	19.531 kHz	39.063 kHz	78.125 kHz
1	0	$f_{MAIN}/2^9$	3.906 kHz	9.766 kHz	19.531 kHz	39.063 kHz
1	1	f_{SUB}	32.768 kHz			

Notes 1. The RMCK102 and RMCK002 bits are only provided in the RMCN102 register. Bits 1 and 0 of the RMCN113 register are read-only and 0 is read.

2. Set this bit when RMENm = 0.

3. RMBDm bit is bit 0 of the RMCN2m register

4. Specify the remote controller receiver source clock by using RMCK102 and RMCK002. The clock selected by using this register is the source clock of all channels. Therefore, set the register when the operation of all channels is stopped (RMENm = 0).

Remarks 1 m = 02, 13

2 f_{MAIN} = 2 to 20 MHz , f_{SUB} = 32.768 kHz

(3) Remote controller receive control 2 register m (RMCN2m)

This register is used to select the remote controller receive operation mode.

RMCN2m is set with a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets RMCN2m to 00H.

Figure 14-9. Format of Remote Controller Receive Control 2 Register m (RMCN2m)

Address: F0331H, F0343H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMCN2m	0	0	0	0	0	RMNCEm	RMEROm	RMBDm

RMNCEm Notes 1, 2	Noise elimination (cyclic elimination) selection
0	Does not perform cyclic elimination (default).
1	Performs cyclic elimination.

RMEROm Notes 1, 3	Selection of operation upon error occurrence
0	Generates error interrupt and clears and restarts bit counter/shift register (default).
1	Generates error interrupt and continues operation, but does not clear bit counter/shift register.

RMBDm Note 1	Remote control receive bit detection method
0	High-/low-width detection (type A/type B/type C) (default)
1	Cycle detection (type B1/type C1)

Notes 1. Set this bit when RMENm = 0.

2. Cyclic elimination is valid with type B1 or C1. The specification of cyclic elimination is ignored with types A, B, and C.

3. Operation continuation upon error occurrence is valid with type B1 or C1. The specification of operation continuation upon error occurrence is ignored with types A, B, and C.

(4) Remote controller receive interrupt status register n (RMINTSn)

This register identifies which interrupt request was generated when a remote control reception interrupt (INTRAn) occurred.

RMINTSn is set with an 8-bit memory manipulation instruction.

Reset signal generation or remote control operation stop (RMENm = 0) clears this register to 00H.

Writing 1 to the target bit clears the target bit.

Figure 14-10. Format of Remote Controller Receive Interrupt Status Register n (RMINTSn)

Address: FFF4DH, FFF57H, FFF5DH, FFF63H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMINTSn	0	0	0	0	0	INTSDFULLn	INTSRENDn	INTSGPn

<Operation when RMINTSn is read>

INTSDFULLn	8-bit data reception completion interrupt request status
0	8-bit data reception completion interrupt has not occurred
1	8-bit data reception completion interrupt has occurred

INTSRENDn	Reception completion interrupt request status
0	Reception completion interrupt has not occurred
1	Reception completion interrupt has occurred

INTSGPn	Guide pulse detection interrupt request status
0	Guide pulse detection interrupt has not occurred
1	Guide pulse detection interrupt has occurred

<Operation when RMINTSn is written>

INTSDFULLn	8-bit data reception completion interrupt request status
0	8-bit data reception completion interrupt status is not cleared
1	8-bit data reception completion interrupt status is cleared

INTSRENDn	Reception completion interrupt request status
0	Reception completion interrupt status is not cleared
1	Reception completion interrupt status is cleared

INTSGPn	Guide pulse detection interrupt request status
0	Guide pulse detection interrupt status is not cleared
1	Guide pulse detection interrupt status is cleared

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

Caution If the interrupt status set timing and the clear timing by this register conflict, the interrupt status set timing is given precedence. Even if the status of the target bit is cleared by using this register, clearing the status may take one source clock (clock set by using RMCK102 and RMCK002, regardless of the setting of RMPRSm).

Remark m = 02, 13, n = 0 to 3

(5) Remote controller receive data slew control register (RMSW)

This register is output control register for the output from ROUT pin of the remote controller reception data input from the RIN01 and RIN23 pins without eliminating noise and without decoding.

RMSW is set with a 1-bit or 8-bit memory manipulation instruction.

Figure 14-11. Format of Remote Controller Receive Data Slew Control Register (RMSW)

Address: F0353H After reset: 00H R/W

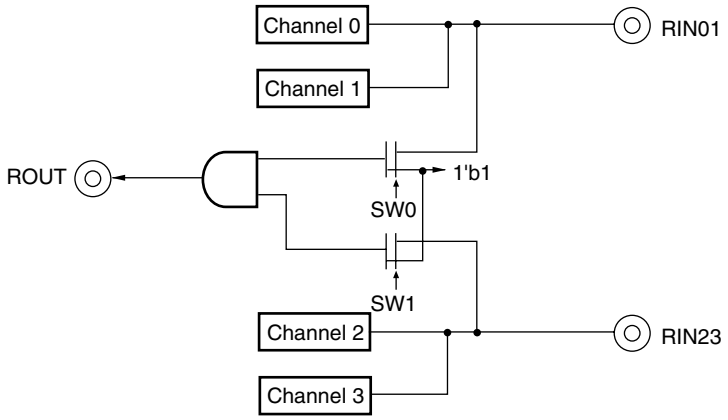
Symbol	7	6	5	4	3	2	1	0
RMSW	0	0	0	0	0	0	RMSW1	RMSW0

RMSW1	Turning SW1 on and off
0	Turn SW1 off (default)
1	Turn SW1 on

RMSW0	Turning SW0 on and off
0	Turn SW0 off (default)
1	Turn SW0 on

The remote controller receive data input from the photo-receiver is input to the AND circuit via SW0 and SW1 and output from a single output pin.

Figure 14-12. Data Slew Switching Circuit



(6) Port mode registers 4 and 9 (PM4, PM9)

These registers set port 4 and 9 input/output in 1-bit units.

When using the P90/ROUT pin for remote controller receive data output, clear PM90 to 0. The output latches of P90 to 1.

When using the P46/RIN01 and P47/RIN23 pins (P46/RIN01/INTP1/TI05/TO05 and P47/RIN23/INTP2 pins in the 78K0R/KG3-C) for remote controller receive data input, set PM46 and PM47 to 1. The output latches of P46 and P47 at this time may be 0 or 1.

PM4 and PM9 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 14-13. Format of Port Mode Registers 4, 9 (PM4, PM9)

Address: FF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

Address: FF29H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM9	1	1	1	1	1	1	PM91	PM90[Note]

PMmn	P4n pin I/O mode selection (m = 4, 9; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note 78K0/KF3-C only

14.4 Compare Registers of Remote Controller Receiver

This register is used to set timing judgment of remote controller reception signal.

(1) Remote controller receive GPLS compare register m (RMGPLSm) (Type B reception mode only)

This register is used to detect the low level of a remote controller guide pulse (short side).

RMGPLSm is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMGPLSm to 00H.

Figure 14-14. Format of Remote Controller Receive GPLS Compare Register m (RMGPLSm)

Address: F0332H, F0344H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMGPLSm								

Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the lower eight bits of the RMGPBSm register for cycle detection (type B1). If the RMBDm bit of the RMCN2m register is 0, this register is valid as the RMGPLSm register. If the RMBDm bit is 1, this register is used as the lower eight bits of RMGPBSm.

Remark m = 02, 13

(2) Remote controller receive GPLL compare register m (RMGPLLm) (Type B reception mode only)

This register is used to detect the low level of a remote controller guide pulse (long side).

RMGPLLm is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMGPLLm to 00H.

Figure 14-15. Format of Remote Controller Receive GPLL Compare Register m (RMGPLLm)

Address: F0333H, F0345H After reset: 00H R/W

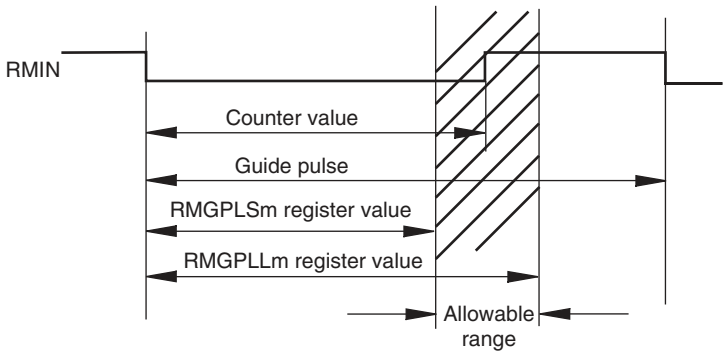
Symbol	7	6	5	4	3	2	1	0
RMGPLLm								

Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the higher eight bits of the RMGPBSm register for cycle detection (type B1). If the RMBDm bit of the RMCN2m register is 0, this register is valid as the RMGPLLm register. If the RMBDm bit is 1, this register is used as the higher eight bits of RMGPBSm.

Remark m = 02, 13

<Allowable range of guide pulse low-level width>



If RMGPLSm ≤ counter value < RMGPLLm is satisfied, it is assumed that the low level of the guide pulse has been successfully received.

If the guide pulse low level is received normally, the guide pulse high-level width will be measured.

Remark m = 02, 13

(3) Remote controller receive GPHS compare register m (RMGPHSm) (Type A or B reception mode only)

This register is used to detect the high level of a remote controller guide pulse (short side).

RMGPHSm is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMGPHSm to 00H.

Figure 14-16. Format of Remote Controller Receive GPHS Compare Register m (RMGPHSm)

Address: F0334H, F0346 After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMGPHSm								

Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the lower eight bits of the RMGPBLm register for cycle detection (type B1). If the RMBDm bit of the RMCN2m register is 0, this register is valid as the RMGPHSm register. If the RMBDm bit is 1, this register is used as the lower eight bits of RMGPBLm.

Remark m = 02, 13

(4) Remote controller receive GPHL compare register m (RMGPHLm) (Type A or B reception mode only)

This register is used to detect the high level of a remote controller guide pulse (long side).

RMGPHLm is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMGPHLm to 00H.

Figure 14-17. Format of Remote Controller Receive GPHL Compare Register m (RMGPHLm)

Address: F0335H, F0347H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMGPHLm								

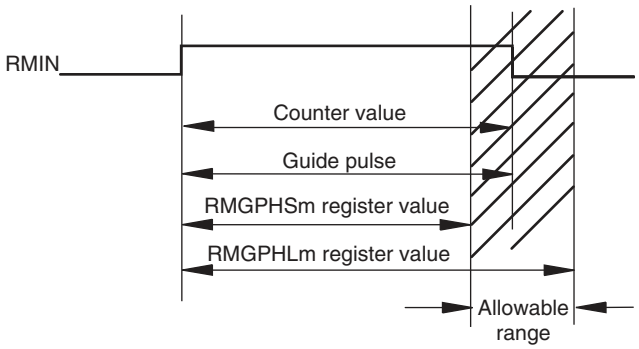
Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the higher eight bits of the RMGPBLm register for cycle detection (type B1). If the RMBDm bit of the RMCN2m register is 0, this register is valid as the RMGPHLm register. If the RMBDm bit is 1, this register is used as the higher eight bits of RMGPBLm.

Remark m = 02, 13

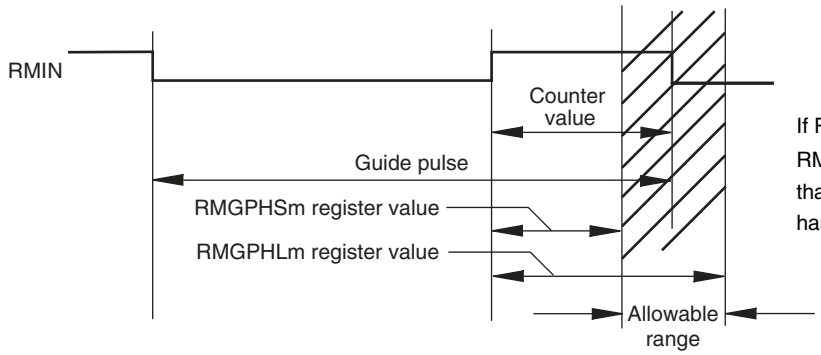
<Allowable range of guide pulse high-level width>

(a) Type A reception mode



If RMGPHSm ≤ counter value < RMGPHLm is satisfied, it is assumed that the high level of the guide pulse has been successfully received.

(b) Type B reception mode



If RMGPHSm ≤ counter value < RMGPHLm is satisfied, it is assumed that the high level of the guide pulse has been successfully received.

If the guide pulse high level is received normally, an INTGPn interrupt signal will be output and the low-level width of the next data will be measured.

Remark m = 02, 13, n = 0 to 3

(5) Remote controller receive DLS compare register m (RMDLSm) (Type A, B, or C reception mode only)

This register is used to detect the low level of a remote controller data (short side).

RMDLSm is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDLSm to 00H.

Figure 14-18. Format of Remote Controller Receive DLS Compare Register m (RMDLSm)

Address: F0336H, F0348H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMDLSm								

Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the lower eight bits of the RMDB0Sm register for cycle detection (type B1/type C1). If the RMBDm bit of the RMCN2m register is 0, this register is valid as the RMDLSm register. If the RMBDm bit is 1, this register is used as the lower eight bits of RMDB0Sm.

Remark m = 02, 13

(6) Remote controller receive DLL compare register m (RMDLLm) (Type A, B, or C reception mode only)

This register is used to detect the low level of a remote controller data (long side).

RMDLLm is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDLLm to 00H.

Figure 14-19. Format of Remote Controller Receive DLL Compare Register m (RMDLLm)

Address: F0337H, F0349H After reset: 00H R/W

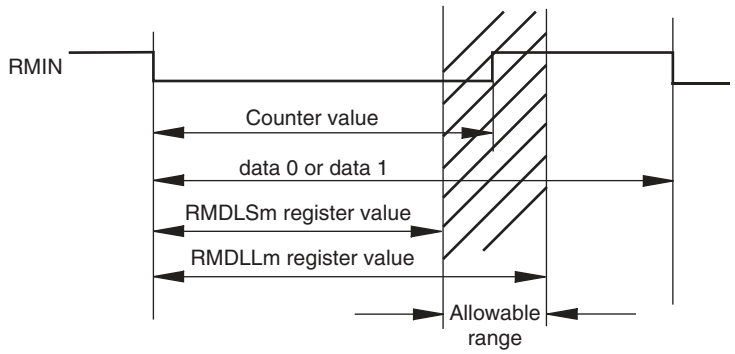
Symbol	7	6	5	4	3	2	1	0
RMDLLm								

Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the higher eight bits of the RMDB0Sm register for cycle detection (type B1/type C1). If the RMBDm bit of the RMCN2m register is 0, this register is valid as the RMDLLm register. If the RMBDm bit is 1, this register is used as the higher eight bits of RMDB0Sm.

Remark m = 02, 13

<Allowable range of data 0 or data 1 low-level width>



If RMDLSm ≤ counter value < RMDLLm is satisfied, it is assumed that the low level of the data 0 or data1 has been successfully received.

If the data low-level is received normally, the data high-level width will be measured.

Remark m = 02, 13

(7) Remote controller receive DH0S compare register m (RMDH0Sm) (Type A, B, or C reception mode only)

This register is used to detect the high level of a remote controller data 0 (short side).

RMDH0Sm is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDH0Sm to 00H.

Figure 14-20. Format of Remote Controller Receive DH0S Compare Register m (RMDH0Sm)

Address: F0338H, F034AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMDH0Sm								

Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the lower eight bits of the RMDB0Lm register for cycle detection (type B1/type C1). If the RMBDm bit of the RMCN2m register is 0, this register is valid as the RMDH0Sm register. If the RMBDm bit is 1, this register is used as the lower eight bits of RMDB0Lm.

Remark m = 02, 13

(8) Remote controller receive DH0L compare register m (RMDH0Lm) (Type A, B, or C reception mode only)

This register is used to detect the high level of a remote controller data 0 (long side).

RMDH0Lm is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDH0Lm to 00H.

Figure 14-21. Format of Remote Controller Receive DH0L Compare Register m (RMDH0Lm)

Address: F0339H, F034BH After reset: 00H R/W

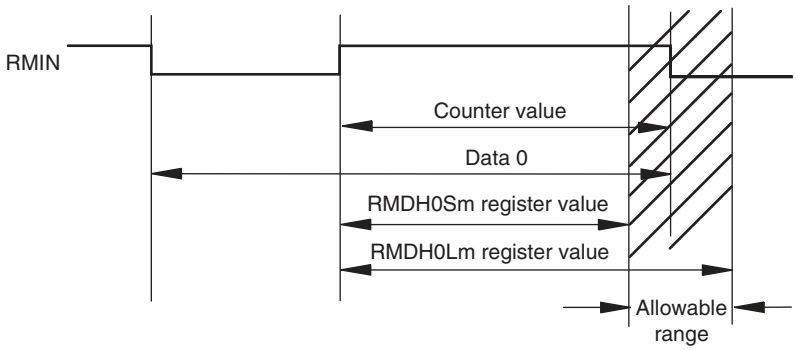
Symbol	7	6	5	4	3	2	1	0
RMDH0Lm								

Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the higher eight bits of the RMDB0Lm register for cycle detection (type B1/type C1). If the RMBDm bit of the RMCN2m register is 0, this register is valid as the RMDH0Lm register. If the RMBDm bit is 1, this register is used as the higher eight bits of RMDB0Lm.

Remark m = 02, 13

<Allowable range of data 0 high-level width>



If RMDH0Sm ≤ counter value < RMDH0Lm is satisfied, it is assumed that the high level of data 0 has been successfully received, and therefore RMSRn receives the data.

If data 0 is received normally, the shift register will be right-shifted, “0” will be stored in bit 7, and the low-level width of the next data will be measured.

Caution Be sure to set the high-level width of data 0 shorter than that of data 1.

Remark m = 02, 13, n = 0 to 3

(9) Remote controller receive DH1S compare register m (RMDH1Sm) (Type A, B, or C reception mode only)

This register is used to detect the high level of remote controller data 1 (short side).

RMDH1Sm is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDH1Sm to 00H.

Figure 14-22. Format of Remote Controller Receive DH1S Compare Register m (RMDH1Sm)

Address: F033AH, F034CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMDH1Sm								

Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the lower eight bits of the RMDB1Sm register for cycle detection (type B1/type C1). If the RMBDm bit of the RMCN2m register is 0, this register is valid as the RMDH1Sm register. If the RMBDm bit is 1, this register is used as the lower eight bits of RMDB1Sm.

Remark m = 02, 13

(10) Remote controller receive DH1L compare register m (RMDH1Lm) (Type A, B, or C reception mode only)

This register is used to detect the high level of remote controller data 1 (long side).

RMDH1Lm is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMDH1Lm to 00H.

Figure 14-23. Format of Remote Controller Receive DH1L Compare Register m (RMDH1Lm)

Address: F033BH, F034DH After reset: 00H R/W

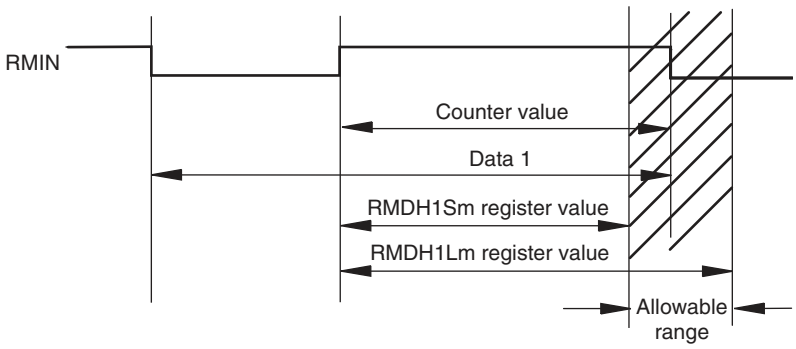
Symbol	7	6	5	4	3	2	1	0
RMDH1Lm								

Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the higher eight bits of the RMDB1Sm register for cycle detection (type B1/type C1). If the RMBDm bit of the RMCN2m register is 0, this register is valid as the RMDH1Lm register. If the RMBDm bit is 1, this register is used as the higher eight bits of RMDB1Sm.

Remark m = 02, 13

<Allowable range of data 1 high-level width>



If RMDH1Sm ≤ counter value < RMDH1Lm is satisfied, it is assumed that the high level of data 1 has been successfully received, and therefore RMSRn receives the data.

If data 1 is received normally, the shift register will be right-shifted, “1” will be stored in bit 7, and the low-level width of the next data will be measured.

Caution Be sure to set the high-level width of data 1 longer than that of data 0.

Remark m = 02, 13, n = 0 to 3

(11) Remote controller receive GPBS compare register m (RMGPBSm) (Type B1 reception mode only)

This register is used to detect the cycle of a remote controller guide pulse (short side).

RMGPBSm is set with a 16-bit memory manipulation instruction.

Reset signal generation sets RMGPBSm to 0000H.

Figure 14-24. Format of Remote Controller Receive GPBS Compare Register m (RMGPBSm)

Address: F0332H, F0344H After reset: 0000H R/W

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

Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the RMGPLSm and RMGPLLm registers for high-/low-level width detection (type B).

To access this register, write this register in 16-bit units after setting the RMBDm bit of the RMCN2m register to 1. If this register is written in 16-bit units while the RMBDm bit is 0, only the lower eight bits will be written, and the higher eight bits will be 0 if this register is read.

Remark m = 02, 13

(12) Remote controller receive GPBL compare register m (RMGPBLm) (Type B1 reception mode only)

This register is used to detect the cycle of a remote controller guide pulse (long side).

RMGPBLm is set with a 16-bit memory manipulation instruction.

Reset signal generation sets RMGPBLm to 0000H.

Figure 14-25. Format of Remote Controller Receive GPBL Compare Register m (RMGPBLm)

Address: F0334H, F0346H After reset: 0000H R/W

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

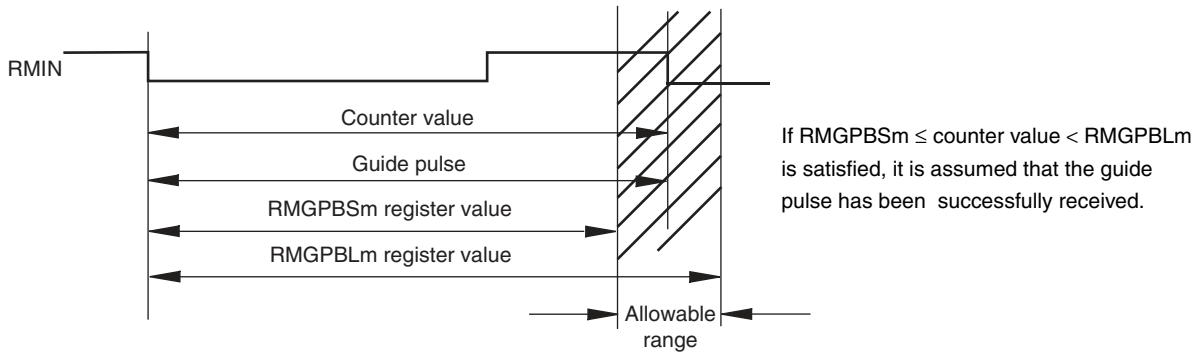
Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the RMGPHSm and RMGPHLm registers for high-/low-level width detection (type A/type B).

To access this register, write this register in 16-bit units after setting the RMBDm bit of the RMCN2m register to 1. If this register is written in 16-bit units while the RMBDm bit is 0, only the lower eight bits will be written, and the higher eight bits will be 0 if this register is read.

Remark m = 02, 13

<Allowable range of guide pulse cycle>



If the guide pulse is received normally, an INTGPn interrupt signal will be output and the cycle of the next data will be measured.

Remark m = 02, 13, n = 0 to 3

(13) Remote controller receive DB0S compare register m (RMDB0Sm) (Type B1 or C1 reception mode only)

This register is used to detect the cycle of a remote controller data 0 (short side).

RMDB0Sm is set with a 16-bit memory manipulation instruction.

Reset signal generation sets RMDB0Sm to 0000H.

Figure 14-26. Format of Remote Controller Receive DB0S Compare Register m (RMDB0Sm)

Address: F0336H, F0348H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RMDB0Sm																

Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the RMDLSm and RMDLLm registers for high-/low-level width detection (type A/type B/type C).

To access this register, write this register in 16-bit units after setting the RMBDm bit of the RMCN2m register to 1. If this register is written in 16-bit units while the RMBDm bit is 0, only the lower eight bits will be written, and the higher eight bits will be 0 if this register is read.

Remark m = 02, 13

(14) Remote controller receive DB0L compare register m (RMDB0Lm) (Type B1 or Type C1 reception mode only)

This register is used to detect the cycle of a remote controller data 0 (long side).

RMDB0Lm is set with a 16-bit memory manipulation instruction.

Reset signal generation sets RMDB0Lm to 0000H.

Figure 14-27. Format of Remote Controller Receive DB0L Compare Register m (RMDB0Lm)

Address: F0338H, F034AH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RMDB0Lm																

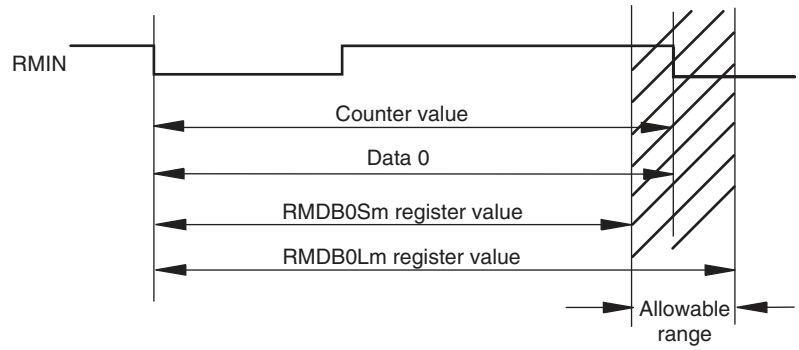
Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the RMDH0Sm and RMDH0Lm registers for high-/low-level width detection (type A/type B/type C).

To access this register, write this register in 16-bit units after setting the RMBDm bit of the RMCN2m register to 1. If this register is written in 16-bit units while the RMBDm bit is 0, only the lower eight bits will be written, and the higher eight bits will be 0 if this register is read.

Remark m = 02, 13

<Allowable range of data 0 cycle>



If RMDB0Sm ≤ counter value < RMDB0Lm is satisfied, it is assumed that the data 0 has been successfully received.

If data 0 is received normally, the shift register will be right-shifted, “0” will be stored in bit 7, and the cycle of the next data will be measured.

Caution Be sure to set the high-level width of data 0 shorter than that of data 1.

Remark m = 02, 13

(15) Remote controller receive DB1S compare register m (RMDB1Sm) (Type B1 or C1 reception mode only)

This register is used to detect the high level of remote controller data 1 (short side).

RMDB1Sm is set with a 16-bit memory manipulation instruction.

Reset signal generation sets RMDB1Sm to 00H.

Figure 14-28. Format of Remote Controller Receive DB1S Compare Register m (RMDB1Sm)

Address: F033AH, F034CH After reset: 0000H R/W

Symbol	7	6	5	4	3	2	1	0
RMDB1Sm								

Cautions 1. Write this register while RMENm = 0.

2. This register is shared with the RMDH1Sm and RMDH1Lm registers for high-/low-level width detection (type A/type B/C type).

To access this register, write this register in 16-bit units after setting the RMBDm bit of the RMCN2m register to 1. If this register is written in 16-bit units while the RMBDm bit is 0, only the lower eight bits will be written, and the higher eight bits will be 0 if this register is read.

Remark m = 02, 13

(16) Remote controller receive DB1L compare register m (RMDB1Lm) (Type B1 or C1 reception mode only)

This register is used to detect the high level of remote controller data 1 (long side).

RMDB1Lm is set with a 16-bit memory manipulation instruction.

Reset signal generation sets RMDB1Lm to 0000H.

Figure 14-29. Format of Remote Controller Receive DB1L Compare Register m (RMDB1Lm)

Address: F033CH, F034EH After reset: 0000H R/W

Symbol	7	6	5	4	3	2	1	0
RMDB1Lm								

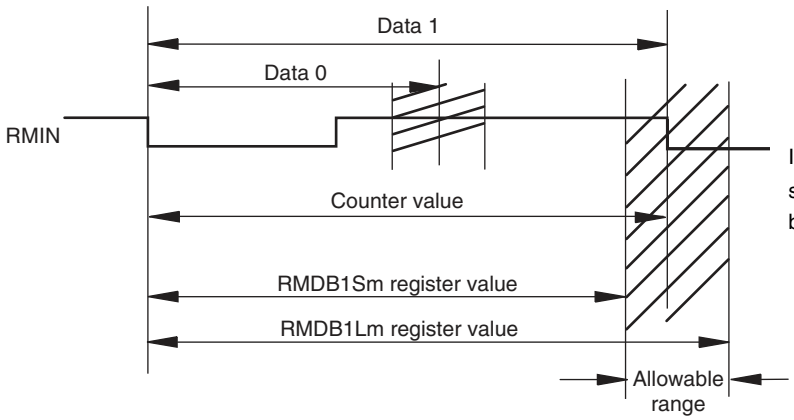
Cautions 1. Write this register while RMENm = 0.

2. This register is not shared with the other registers.

To access this register, write this register in 16-bit units after setting the RMBDm bit of the RMCN2m register to 1. If this register is written in 16-bit units while the RMBDm bit is 0, only the lower eight bits will be written, and the higher eight bits will be 0 if this register is read.

Remark m = 02, 13

<Allowable range of data 1 cycle>



If data 1 is received normally, the shift register will be right-shifted, “1” will be stored in bit 7, and the cycle of the next data will be measured.

Caution Be sure to set the high-level width of data 1 longer than that of data 0.

Remark m = 02, 13

(17) Remote controller receive end width select register m (RMERHm and RMERLm)

This register determines the interval between the timing at which the INTRENDn signal is output.

The conditions for entering the end state differ for each reception mode.

This register functions as a 16-bit register (RMERm) by combining the higher eight bits of RMERHm and the lower eight bits of RMERLm.

RMERHm and RMERLm are set by 8-bit memory manipulation instructions.

Reset signal generation sets RMERHm and RMERLm to 00H.

Figure 14-30. Format of Remote Controller Receive End width Select Register m (RMERHm and RMERLm)

Address: F033EH, F033FH, F0350H, F0351H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMERHm								
RMERLm								

Cautions 1. The same register is used for measuring the end width with types A, B, and C (high-/low-level width detection) and types B1 and C1 (cycle detection).

2. Write a value while RMENm = 0.

$$RMERm = \left\{ \frac{Twe \times (1 - X\%)}{1 / f_{REMPRS}} \right\} - 1$$

Twe: End time

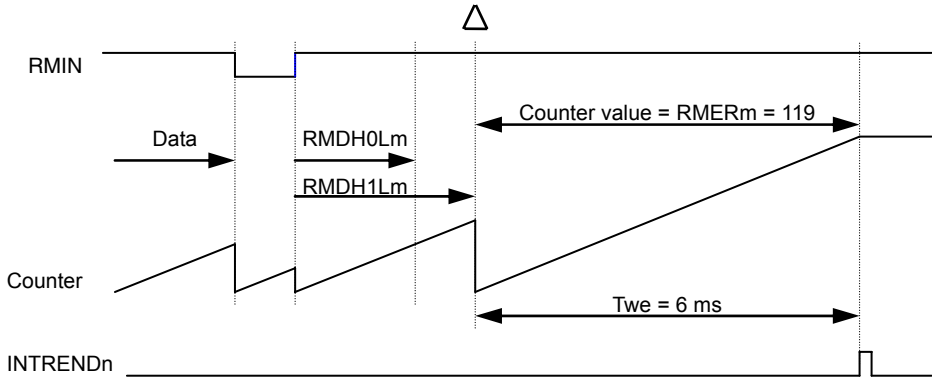
$1/f_{REMPRS}$: Internal operation clock cycle value after frequency division control

X: Allowable error

{ }: Truncate the fractional part of the value calculated by the expression in the brackets.

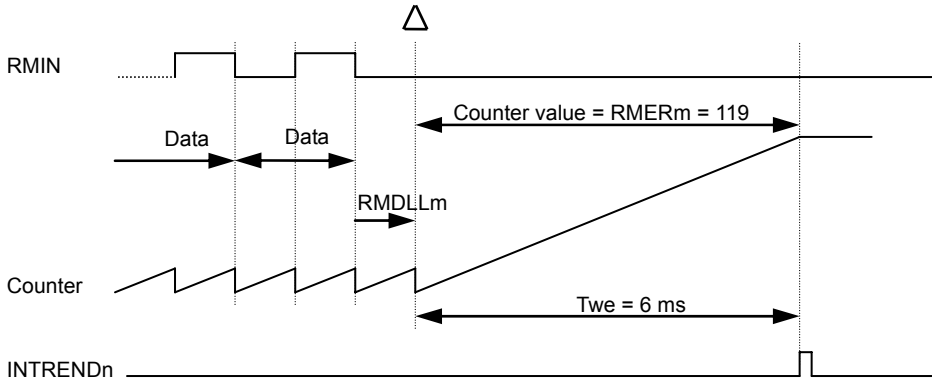
Example 1: With type B or C, if the provisional end time is 6 ms and the internal operation clock cycle is 50 μs, the RMERm register value will be {6 ms/50 μs} − 1 = 119.

A length of (RMDH1Lm + 1 + RMERm + 1) x 50 μs, however, will be required as the high-level width of RMIN. If this cannot be met, an error interrupt (INTRERRn) will be generated.

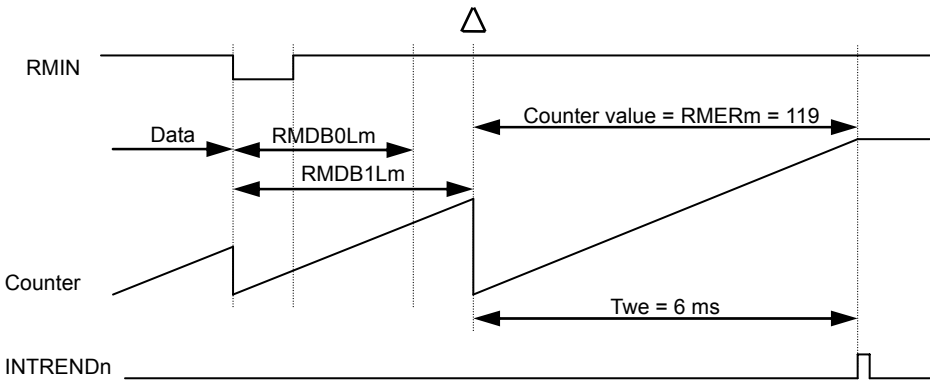


Remark m = 02, 13, n = 0 to 3

Example 2: With type A, if the provisional end time is 6 ms and the internal operation clock cycle is 50 μs, the RMERm register value will be {6 ms/50 μs} – 1 = 119.
A length of (RMDLLm + 1 + RMERm + 1) x 50 μs, however, will be required as the low-level width of RMIN. If this cannot be met, an error interrupt (INTRERRn) will be generated.



Example 3: With type B1 or C1, if the provisional end time is 6 ms and the internal operation clock cycle is 50 μs, the RMERm register value will be {6 ms/50 μs} – 1 = 119.
A length of (RMDB1Lm + 1 + RMERm + 1) x 50 μs, however, will be required as the cycle of RMIN. If this cannot be met, an error interrupt (INTRERRn) will be generated.



Remark m = 02, 13, n = 0 to 3

(18) Remote controller receive noise elimination period setting register m (RMNCP1m) (type B1 or C1 reception modes only)

This register sets the remote control reception noise elimination period.

It specifies the elimination cycle when noise cycle elimination is specified (RMNCEm of the RMCN2m register = 1).

It ignores the fall of the $\overline{\text{RIN}}$ input with the specified elimination cycle.

RMNCP1m is set with an 8-bit memory manipulation instruction.

Reset signal generation sets RMNCP1m to 00H.

Figure 14-31. Format of Remote Controller Receive Noise Elimination Period Setting Register m (RMNCP1m)

Address: F0340H, F0352H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMNCP1m								

Cautions 1. Write a value while RMENm = 0.

2. The noise elimination cycle is counted by the internal operation clock after frequency division control (f_{REMPRS}).

Remark m = 02, 13

14.5 Operation of Remote Controller Receiver

The following remote controller reception mode is used for this remote controller receiver.

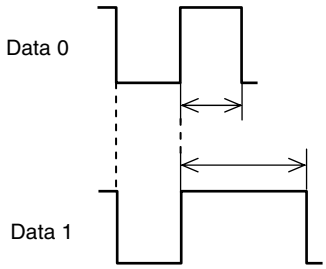
Reception mode	Guide Pulse Low-Level Period	Guide Pulse High-Level Period	Input Waveform to RMIN [Note]	Detection method
Type A	Not used	Used	RIN	High/low level width
Type B	Used	Used	$\overline{\text{RIN}}$	
Type C	Not used	Not used		
Type B1	Used	Used		Cycle
Type C1	Not used	Not used		

Note Refer to **Figure 14-1**.

The data waveforms are as follows.

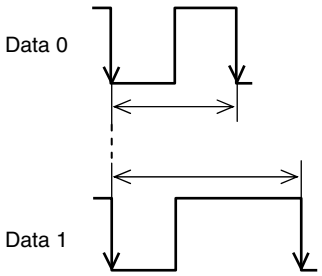
- In type A, type B, and type C reception mode,

  Data 0 and data 1 are identified by the length of the data high-level width (2 types), with the data low-level width fixed.



- In type B1 and type C1 reception mode,

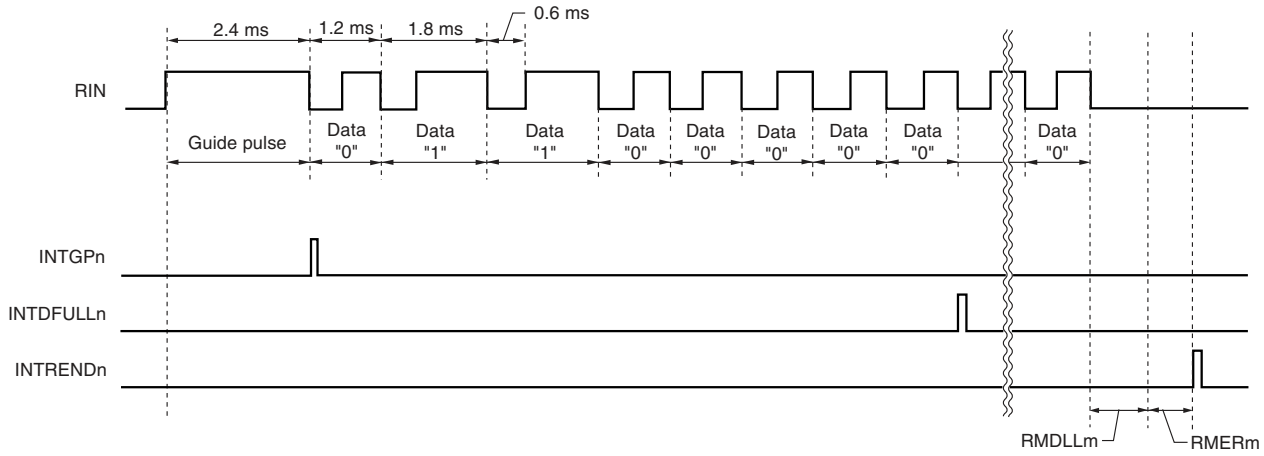
  Data 0 and data 1 are identified by the cycle of the falling edge.



14.5.1 Format of type A reception mode

Figure 14-32 shows the data format for type A.

Figure 14-32. Example of Type A Data Format



Remark m = 02, 13, n = 0 to 3

14.5.2 Operation flow of type A reception mode

Figure 14-33 shows the operation flow.

Cautions 1. When INTRERRn is generated, RMSRn and RMSCRn are automatically cleared immediately.

2. When data has been set to all the bits of RMSRn, the following processing is automatically performed.

- **The value of RMSRn is transferred to RMDRn.**
- **INTDFULLn is generated.**
- **RMSRn is cleared.**

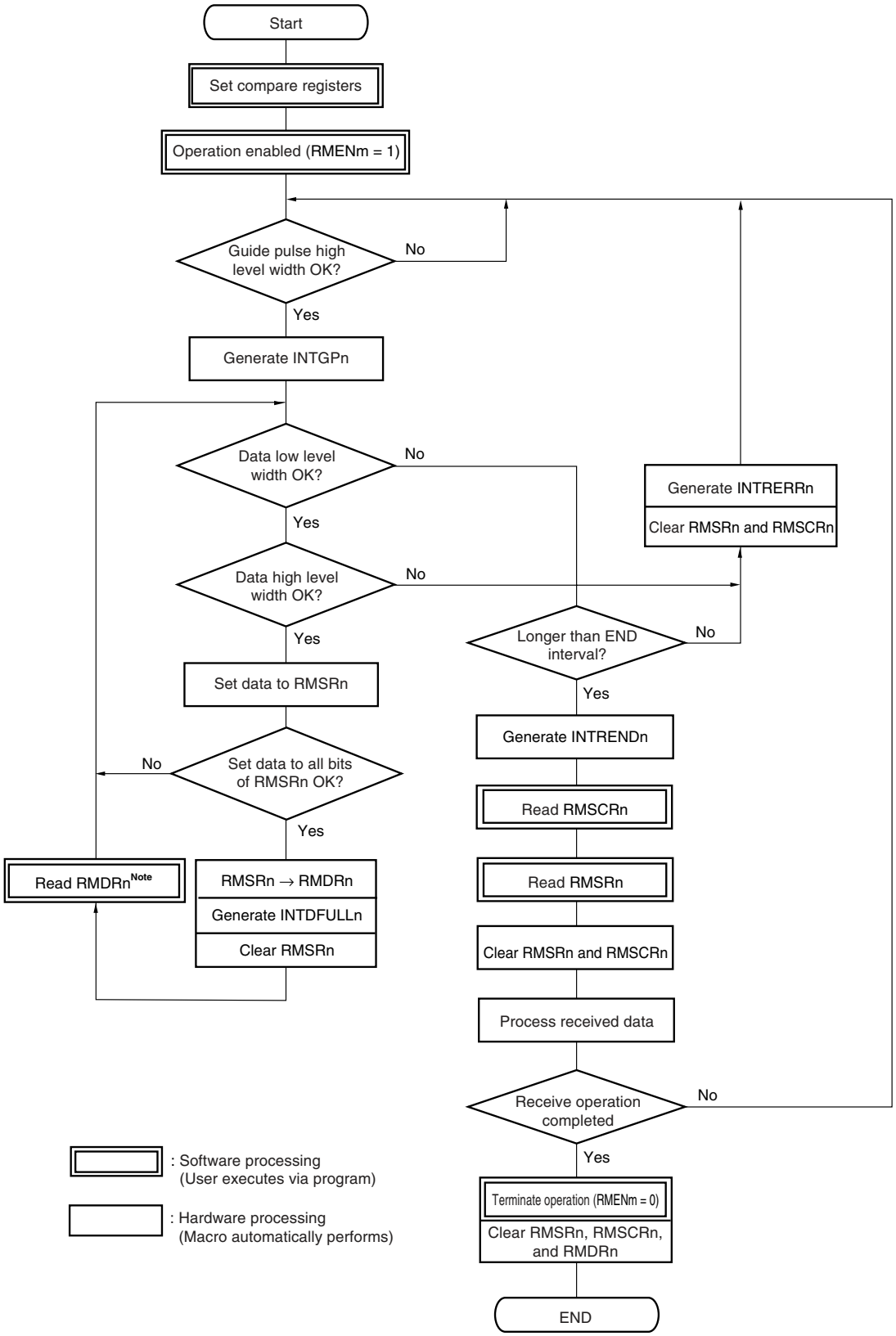
RMDRn must then be read before the next data is set to all the bits of RMSRn.

3. When INTRENDn has been generated, read RMSCRn first followed by RMSRn.
When RMSRn has been read, RMSCRn and RMSRn are automatically cleared.
If INTRENDn is generated, the next data cannot be received until RMSRn is read.

4. RMSRn, RMSCRn, and RMDRn are cleared simultaneously to operation termination (RMENm = 0).

Remark m = 02, 13, n = 0 to 3

Figure 14-33. Operation Flow of Type A Reception Mode



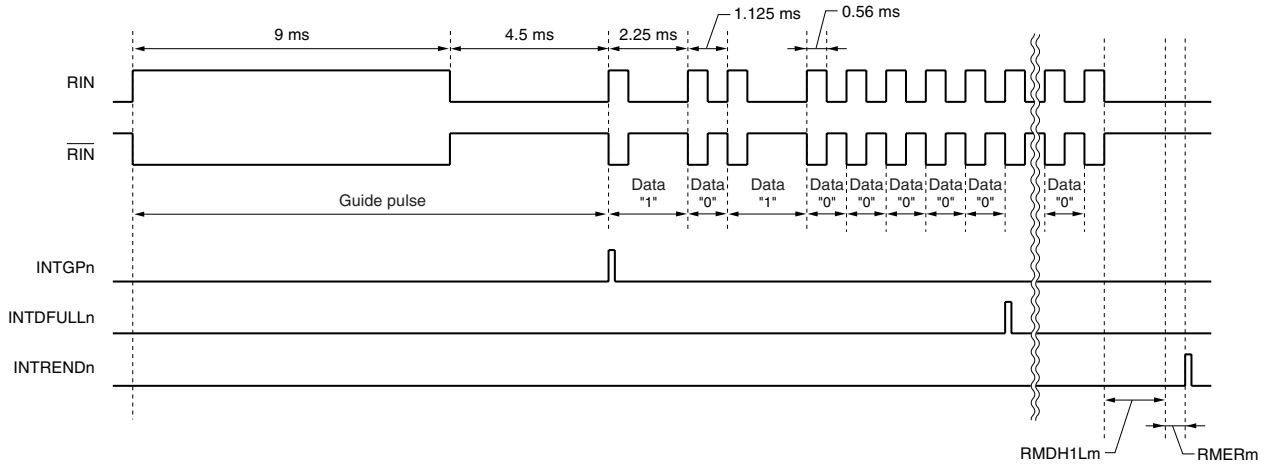
Note Read RMDRn before data has been set to all the bits of RMSRn.

Remark m = 02, 13, n = 0 to 3

14.5.3 Format of type B reception mode

Figure 14-34 shows the data format for type B.

Figure 14-34. Example of Type B Data Format



Remarks 1. m = 02, 13, n = 0 to 3

2. $\overline{\text{RIN}}$ is the internally inverted signal of RIN. Input the $\overline{\text{RIN}}$ waveform to RMIN (see **Figure 14-1**).

14.5.4 Operation flow of type B reception mode

Figure 14-35 shows the operation flow.

Cautions 1. When INTRERRn is generated, RMSRn and RMSCRn are automatically cleared immediately.

2. When data has been set to all the bits of RMSRn, the following processing is automatically performed.

- **The value of RMSRn is transferred to RMDRn.**
- **INTDFULLn is generated.**
- **RMSRn is cleared.**

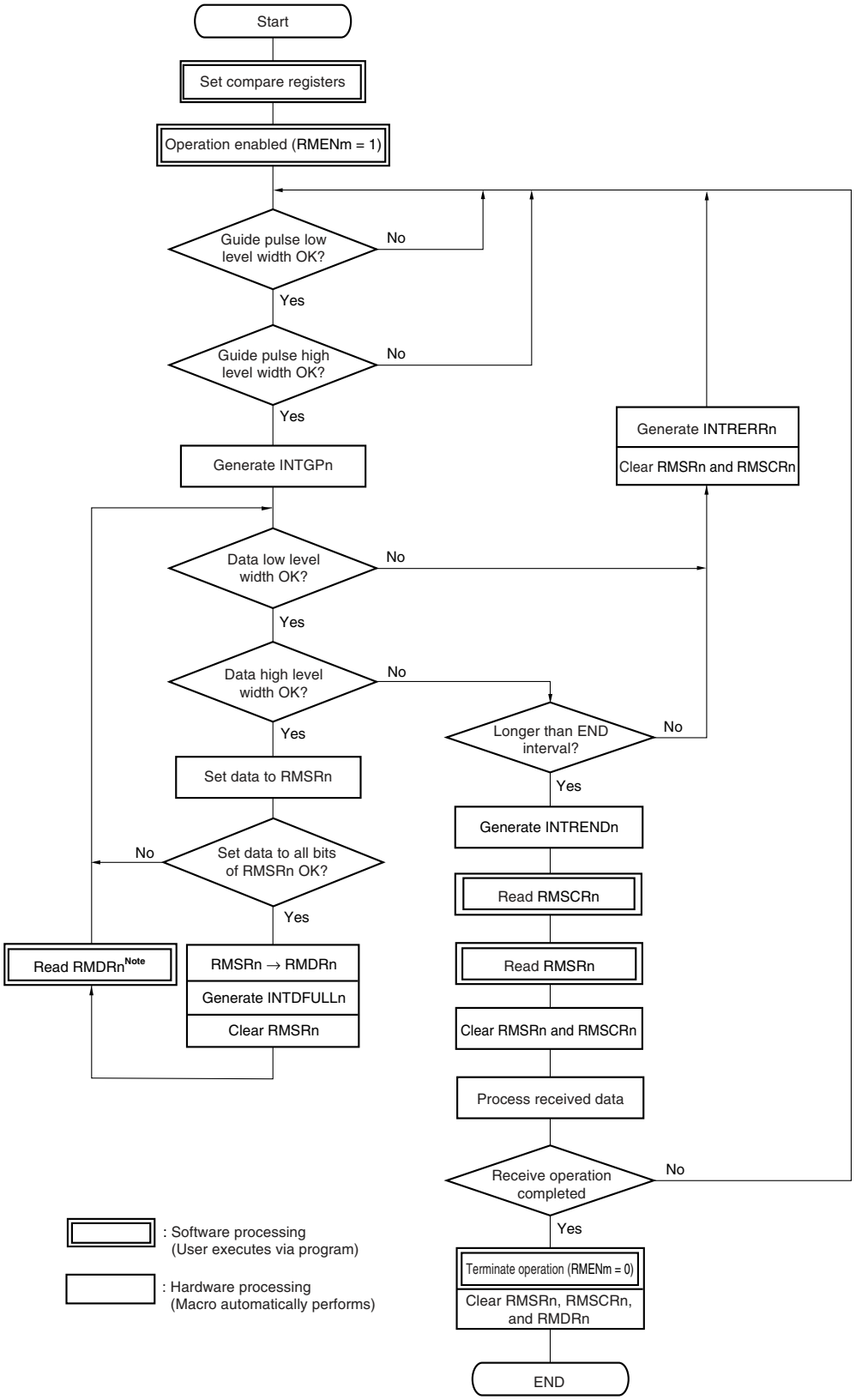
RMDRn must then be read before the next data is set to all the bits of RMSRn.

3. When INTRENDn has been generated, read RMSCRn first followed by RMSRn.
When RMSRn has been read, RMSCRn and RMSRn are automatically cleared.
If INTRENDn is generated, the next data cannot be received until RMSRn is read.

4. RMSRn, RMSCRn, and RMDRn are cleared simultaneously to operation termination (RMENm = 0).

Remark m = 02, 13, n = 0 to 3

Figure 14-35. Operation Flow of Type B Reception Mode



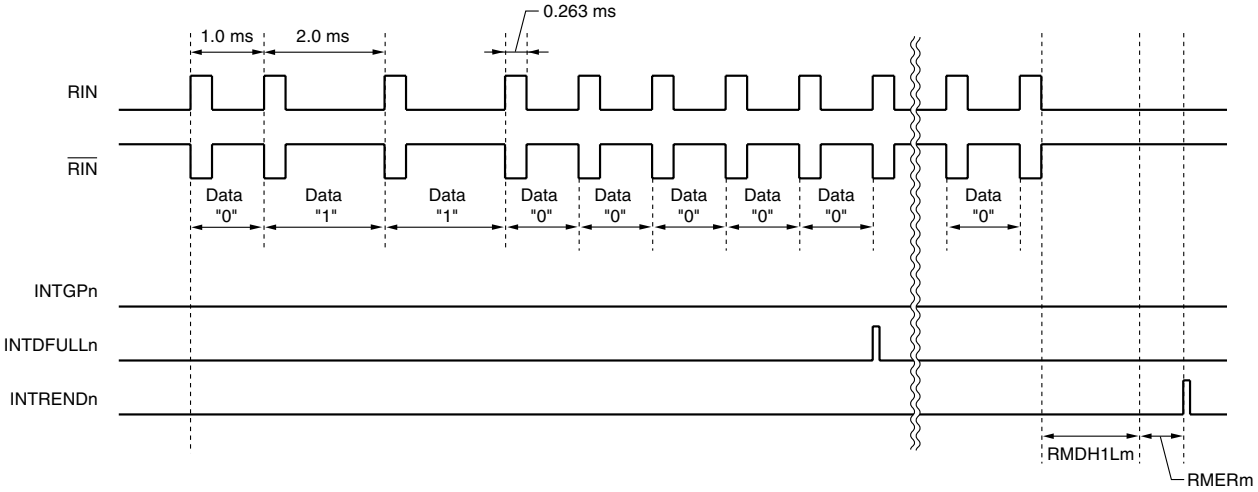
Note Read RMDRn before data has been set to all the bits of RMSRn.

Remark m = 02, 13, n = 0 to 3

14.5.5 Format of type C reception mode

Figure 14-36 shows the data format for type C.

Figure 14-36. Example of Type C Data Format



Remarks 1. m = 02, 13, n = 0 to 3

2. $\overline{\text{RIN}}$ is the internally inverted signal of RIN. Input the $\overline{\text{RIN}}$ waveform to RMIN (see **Figure 14-1**).

14.5.6 Operation flow of type C reception mode

Figure 14-37 shows the operation flow.

Cautions 1. When INTRERRn is generated, RMSRn and RMSCRn are automatically cleared immediately.

2. When data has been set to all the bits of RMSRn, the following processing is automatically performed.

- **The value of RMSRn is transferred to RMDRn.**
- **INTDFULLn is generated.**
- **RMSRn is cleared.**

RMDRn must then be read before the next data is set to all the bits of RMSRn.

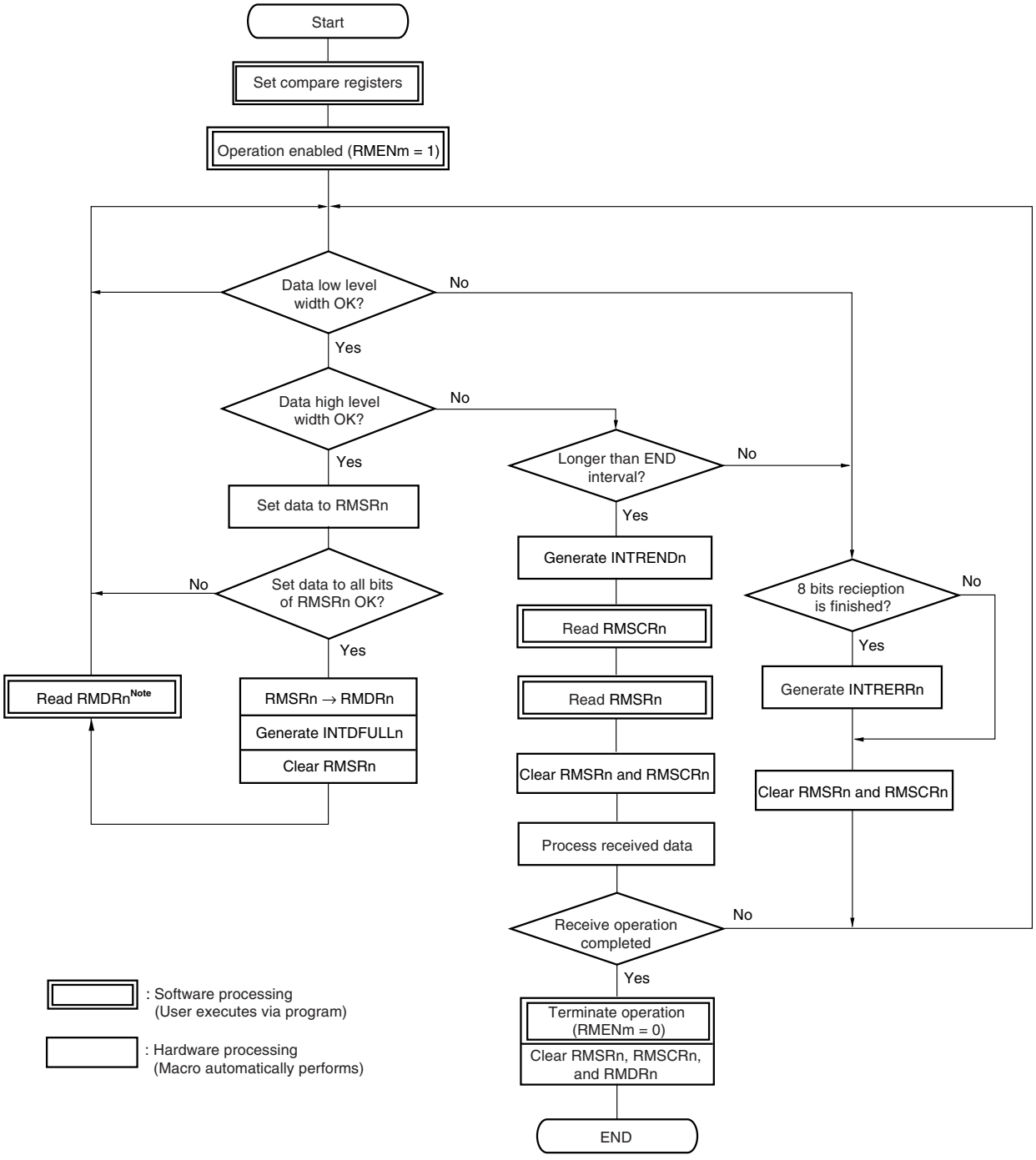
3. When INTRENDn has been generated, read RMSCRn first followed by RMSRn.
When RMSRn has been read, RMSCRn and RMSRn are automatically cleared.
If INTRENDn is generated, the next data cannot be received until RMSRn is read.

4. RMSRn, RMSCRn, and RMDRn are cleared simultaneously to operation termination (RMENm = 0).

5. In type C reception mode, if the conditions for receiving a data low-/high-level width are not met before the first INTDFULLn interrupt is generated, INTRERRn will not be generated. However, RMSRn and RMSCRn will be cleared.

Remark m = 02, 13, n = 0 to 3

Figure 14-37. Operation Flow of Type C Reception Mode



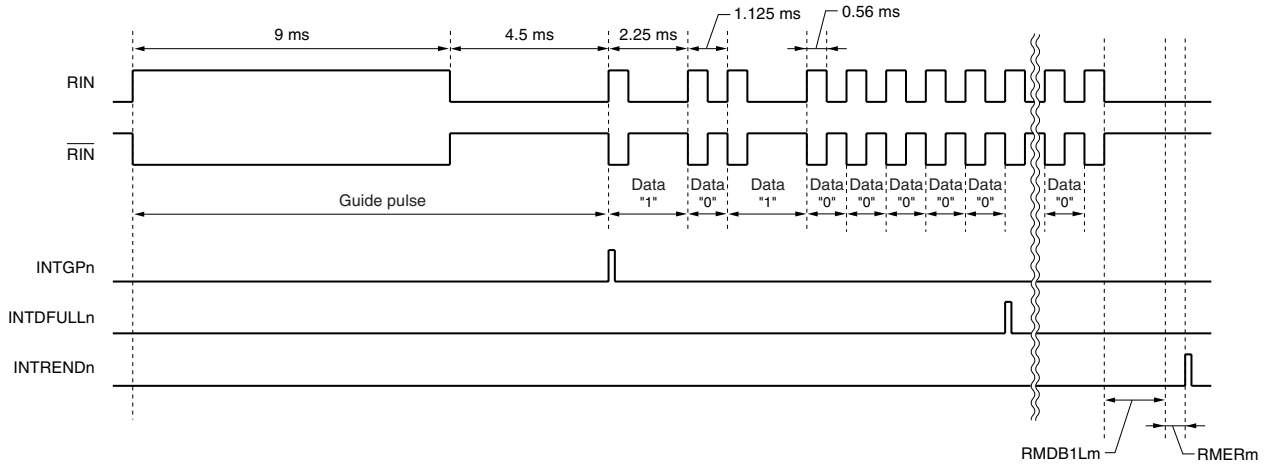
Note Read RMDRn before data has been set to all the bits of RMSRn.

Remark m = 02, 13, n = 0 to 3

14.5.7 Format of type B1 reception mode

Figure 14-38 shows the data format for type B1.

Figure 14-38. Example of Type B1 Data Format



Remarks 1. m = 02, 13, n = 0 to 3

2. $\overline{\text{RIN}}$ is the internally inverted signal of RIN. Input the $\overline{\text{RIN}}$ waveform to RMIN (see **Figure 14-1**).

14.5.8 Operation flow of type B1 reception mode

Figure 14-39 shows the operation flow.

Cautions 1. When data has been set to all the bits of RMSRn, the following processing is automatically performed.

- **The value of RMSRn is transferred to RMDRn.**
- **The value of RMERBSRn is transferred to RMERBDn.**
- **INTDFULLn is generated.**
- **RMSRn is cleared.**

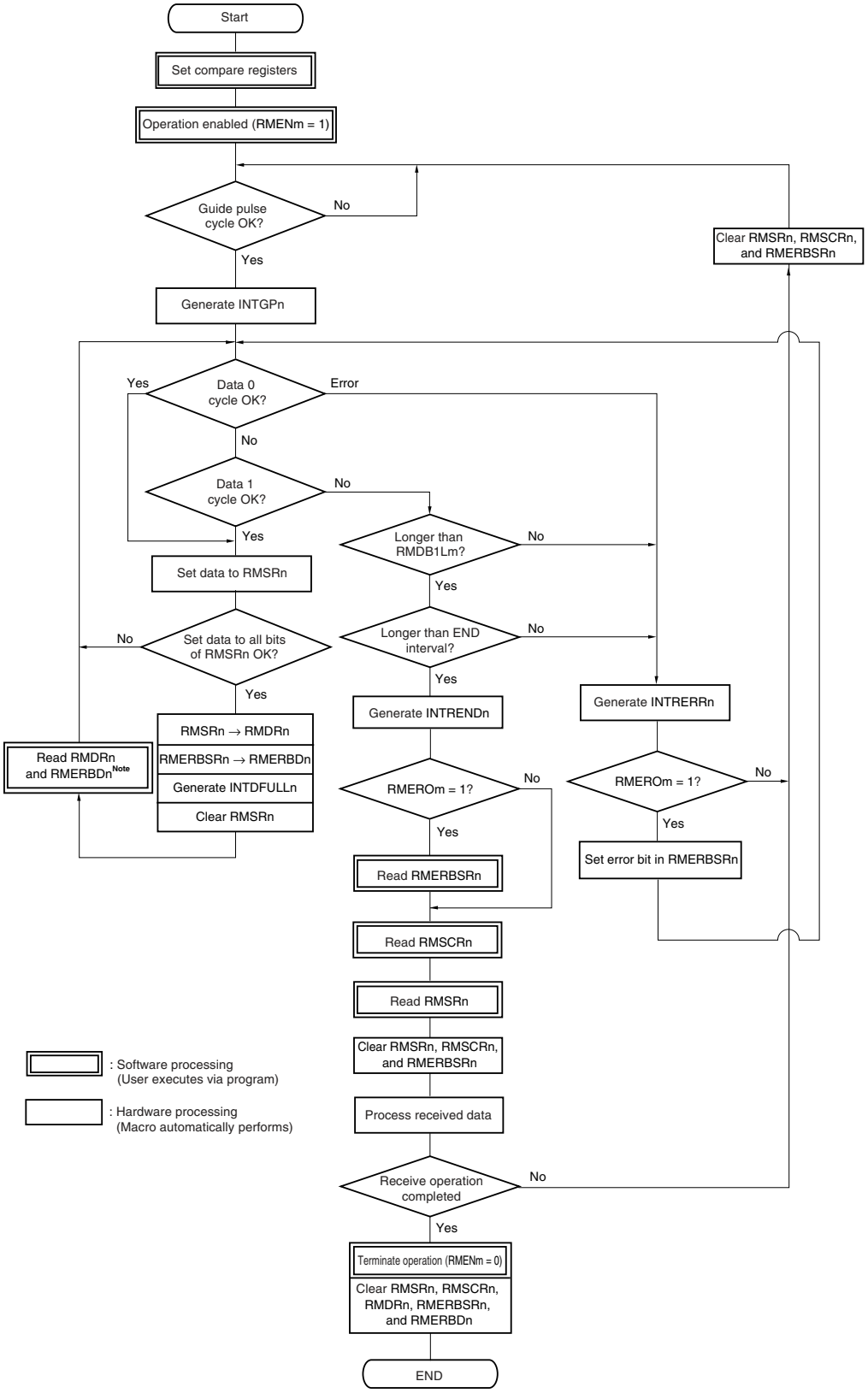
RMDRn and RMERBDn must then be read before the next data is set to all the bits of RMSRn.

2. When RMEROm = 0, read RMSCRn first followed by RMSRn after INTRENDn has been generated.
When RMEROm = 1, read RMSCRn and RMERBSRn first followed by RMSRn after INTRENDn has been generated.
When RMSRn has been read, RMSCRn, RMERBSRn, and RMSRn are automatically cleared.
If INTRENDn is generated, the next data cannot be received until RMSRn is read.

3. RMSRn, RMSCRn, RMDRn, RMERBSRn, and RMERBDn are cleared simultaneously to operation termination (RMENm = 0).

Remark m = 02, 13, n = 0 to 3

Figure 14-39. Operation Flow of Type B1 Reception Mode



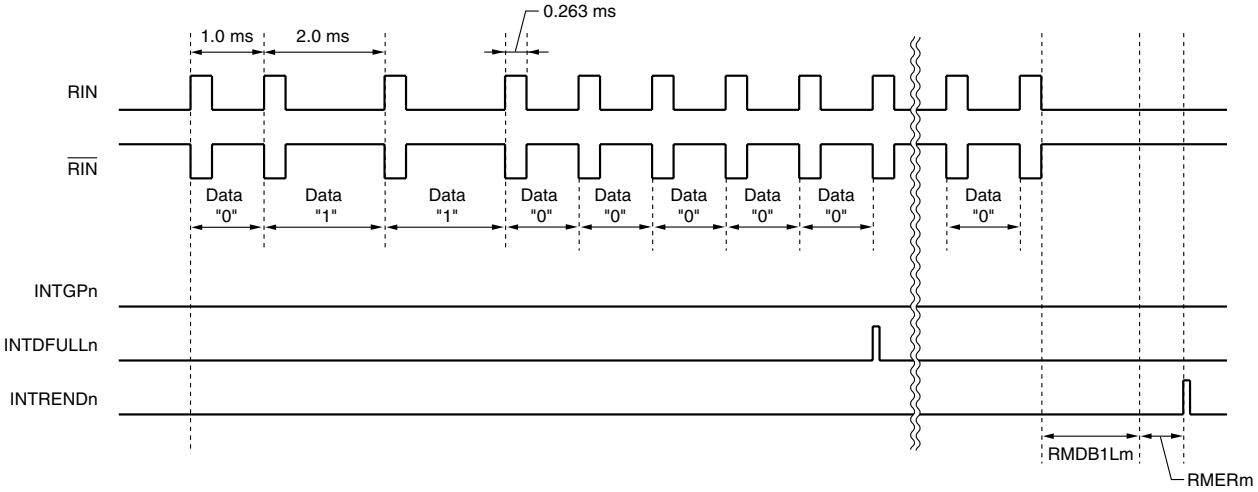
Note Read RMDRn and RMERBDn before data has been set to all the bits of RMSRn.

Remark m = 02, 13, n = 0 to 3

14.5.9 Format of type C1 reception mode

Figure 14-40 shows the data format for type C.

Figure 14-40. Example of Type C1 Data Format



Remarks 1. m = 02, 13, n = 0 to 3

2. $\overline{\text{RIN}}$ is the internally inverted signal of RIN. Input the $\overline{\text{RIN}}$ waveform to RMIN (see **Figure 14-1**).

14.5.10 Operation flow of type C1 reception mode

Figure 14-41 shows the operation flow.

Cautions 1. When data has been set to all the bits of RMSRn, the following processing is automatically performed.

- **The value of RMSRn is transferred to RMDRn.**
- **The value of RMERBSRn is transferred to RMERBDn.**
- **INTDFULLn is generated.**
- **RMSRn is cleared.**

RMDRn and RMERBDn must then be read before the next data is set to all the bits of RMSRn.

2. When RMEROm = 0, read RMSCRn first followed by RMSRn after INTRENDn has been generated.
When RMEROm = 1, read RMSCRn and RMERBSRn first followed by RMSRn after INTRENDn has been generated.
When RMSRn has been read, RMSCRn, RMERBSRn, and RMSRn are automatically cleared.
If INTRENDn is generated, the next data cannot be received until RMSRn is read.

3. RMSRn, RMSCRn, RMDRn, RMERBSRn, and RMERBDn are cleared simultaneously to operation termination (RMENm = 0).

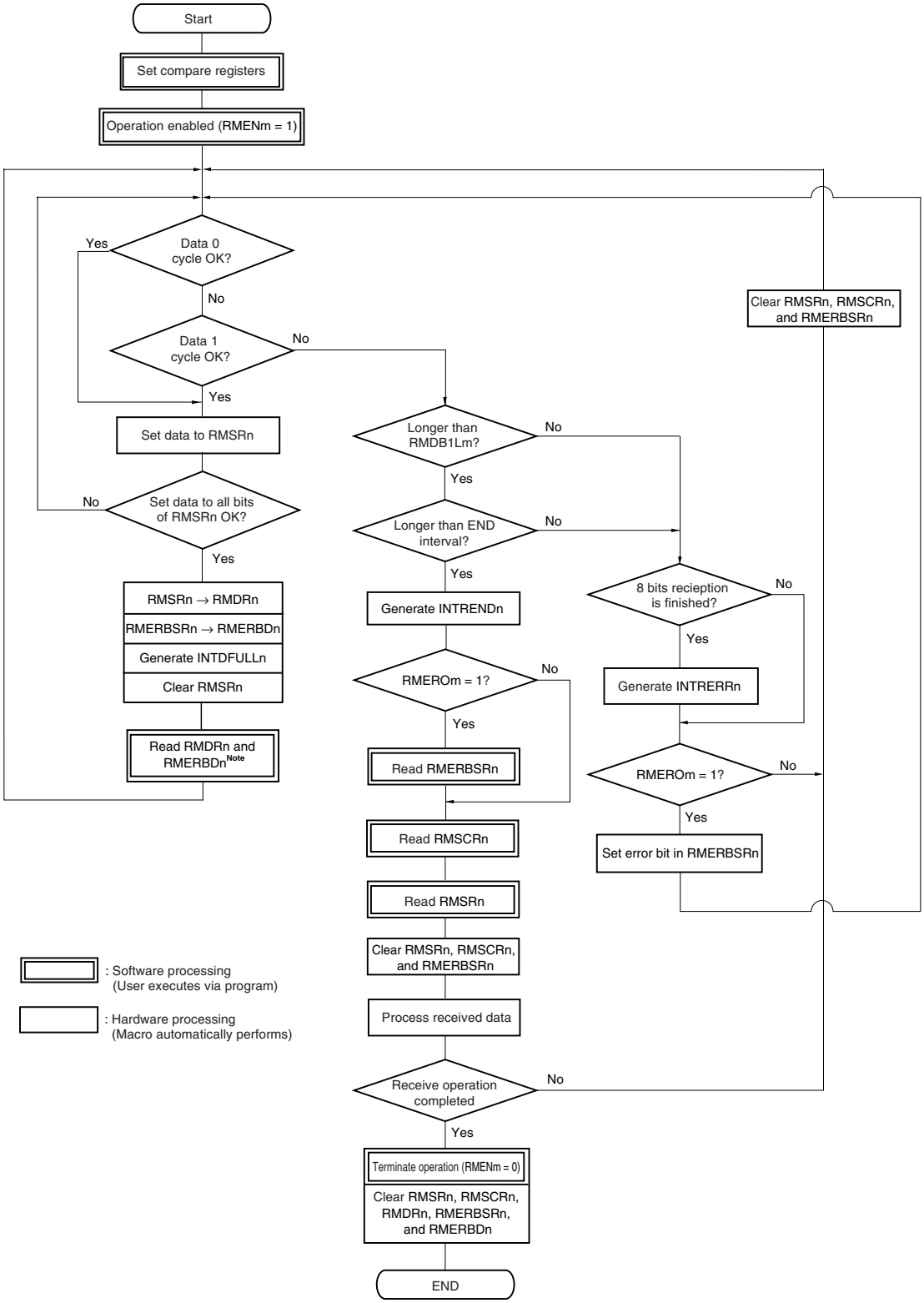
4. In type C1 reception mode, if an error occurs before the first INTDFULLn interrupt is generated, INTRERRn will not be generated.

5. When an error occurs, the processing of RMSRn and RMSCRn depends on the RMEROm bit.

- **If RMEROm = 0, RMSRn and RMSCRn will be cleared when the error occurs.**
- **If RMEROm = 1, RMSRn and RMSCRn will not be cleared when the error occurs.**

Remark m = 02, 13, n = 0 to 3

Figure 14-41. Operation Flow of Type C1 Reception Mode



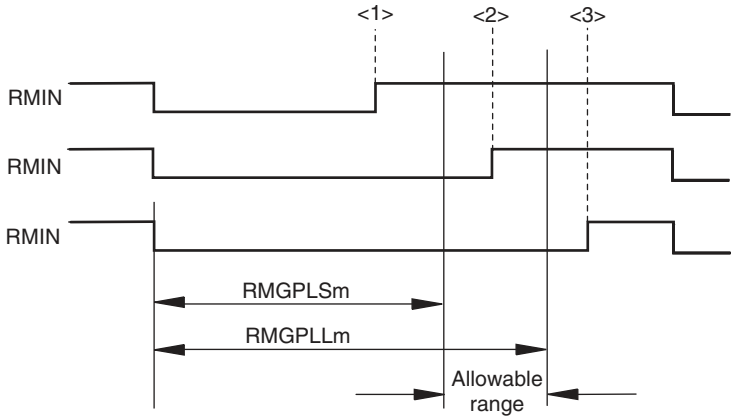
Note Read RMDRn and RMERBDn before data has been set to all the bits of RMSRn.

Remark m = 02, 13, n = 0 to 3

14.5.11 Timing

Operation varies depending on the positions of the RMIN input waveform below.

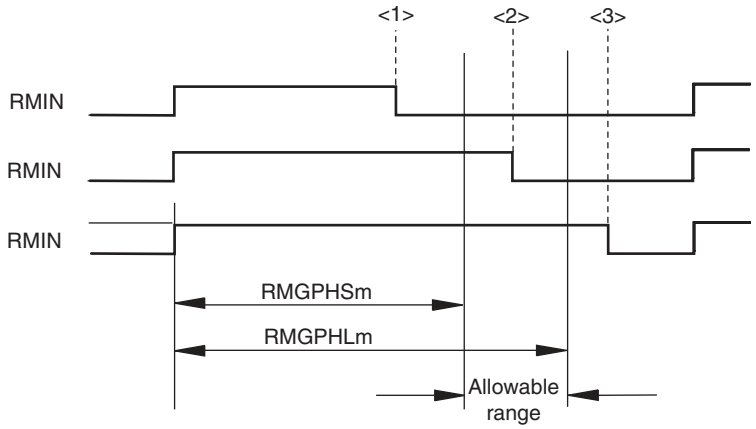
(1) Guide pulse low level width determination (Type B reception mode only)



Condition		Determination	Subsequent operation
<1>	Counter < RMGPLSm	NG (Short)	Detecting the guide pulse continues. Measuring guide pulse low-level width is started from the next falling edge.
<2>	RMGPLSm ≤ counter < RMGPLLm	OK (Within the range)	Guide pulse high-level width measurement is started.
<3>	RMGPLLm ≤ counter	NG (Long)	Detecting the guide pulse continues. Measuring guide pulse low-level width is started from the next falling edge.

Remark m = 02, 13

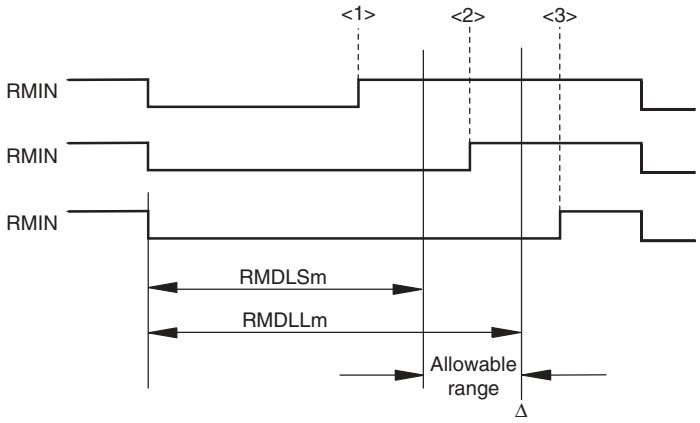
(2) Guide pulse high level width determination (Type A or Type B reception modes only)



Condition		Determination	Subsequent operation
<1>	Counter < RMGPHSm	NG (Short)	(Type A reception mode) Detecting the guide pulse continues. Measuring guide pulse high-level width is started from the next rising edge. (Type B reception modes) Detecting the guide pulse continues and operation restarts from determining the low-level width of the guide pulse. Measuring guide pulse low-level width is started from this falling edge.
<2>	RMGPHSm ≤ counter < RMGPHLm	OK (Within the range)	INTGPn is generated. Data measurement is started.
<3>	RMGPHLm ≤ counter	NG (Long)	(Type A reception mode) Detecting the guide pulse continues. Measuring guide pulse high-level width is started from the next rising edge. (Type B reception modes) Detecting the guide pulse continues and operation restarts from determining the low-level width of the guide pulse. Measuring guide pulse low-level width is started from this falling edge.

Remark m = 02, 13, n = 0 to 3

(3) Data low level width determination (Type A, B, or C reception modes only)

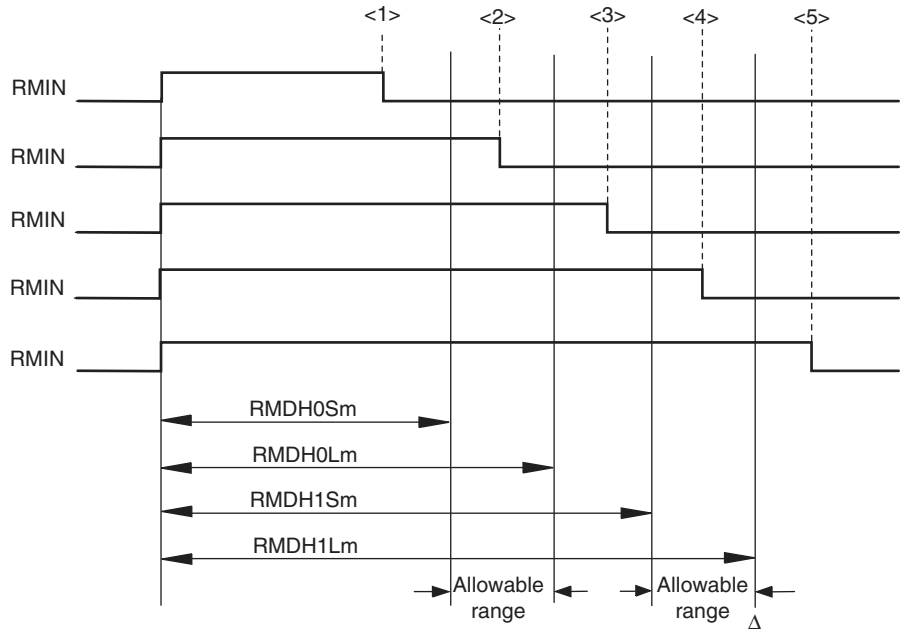


Condition		Determination	Subsequent operation
<1>	Counter < RMDLSm	NG (Short)	Error interrupt INTRERRn is generated Note. (Type A reception mode) Detecting the guide pulse starts. Measuring guide pulse low-level width is started from this rising edge. (Type B reception mode) Detecting the guide pulse starts and operation restarts from determining the low-level width of the guide pulse. Measuring guide pulse low-level width is started from the next falling edge. (Type C reception mode) Detecting the data continues and operation restarts from determining the data low-level width. Measuring data low-level width is started from the next falling edge.
<2>	RMDLSm ≤ counter < RMDLLm	OK (Within the range)	Measuring data high-level width is started.
<3>	RMDLLm ≤ counter	OK (Type A) / NG (Type B, Type C)	(Type A reception mode) Measuring the end width is started from the Δ point. (Type B reception mode) Error interrupt INTRERRn is generated at the Δ point. Detecting the guide pulse starts and operation restarts from determining the low-level width of the guide pulse. Measuring guide pulse low-level width is started from the next falling edge. (Type C reception modes) Error interrupt INTRERRn is generated at the Δ point Note. Detecting the data continues. Measuring data low-level width is started from the next falling edge.

Note In type C reception mode, before the first INTDFULLn interrupt is generated, INTRERRn will not be generated. However, RMSRn and RMSCRn will be cleared.

Remark m = 02, 13, n = 0 to 3

(4) Data high level width determination (Type A, B, or C reception modes only)



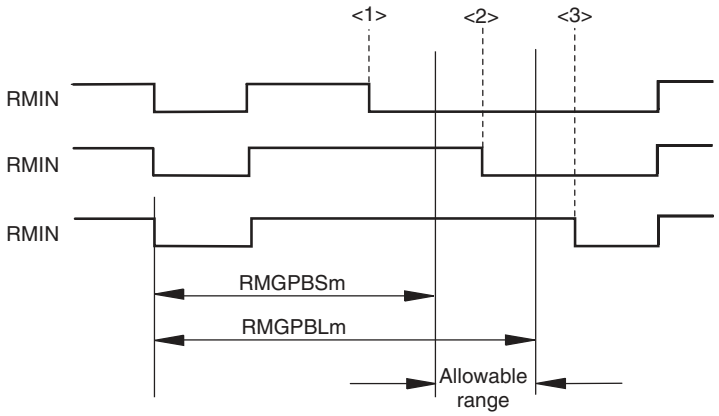
Remark m = 02, 13

Condition		Determination	Subsequent operation
<1>	Counter < RMDH0Sm	NG (Short)	Error interrupt INTRERRn is generated Note. (Type A reception mode) Detecting the guide pulse starts. Measuring guide pulse high-level width is started at the next rising edge. (Type B reception mode) Detecting the guide pulse starts. Measuring guide pulse low-level width is started from this falling edge. (Type C reception mode) Detecting the data continues and operation restarts from determining the data low-level width. Measuring data low-level width is started from this falling edge.
<2>	RMDH0Sm ≤ counter < RMDH0Lm	OK (Within the range)	Data 0 is received. Detecting the data continues. Measuring data low-level width is started from this falling edge.
<3>	RMDH0Lm ≤ counter < RMDH1Sm	NG (Outside of the range)	Error interrupt INTRERRn is generated Note. (Type A reception mode) Detecting the guide pulse starts. Measuring guide pulse high-level width is started at the next rising edge. (Type B reception mode) Detecting the guide pulse starts. Measuring guide pulse low-level width is started from this falling edge. (Type C reception mode) Detecting the data continues and operation restarts from determining the data low-level width. Measuring data low-level width is started from this falling edge.
<4>	RMDH1Sm ≤ counter < RMDH1Lm	OK (Within the range)	Data 1 is received. Detecting the data continues. Measuring the data low-level width is started from this falling edge.
<5>	RMDH1Lm ≤ counter	NG (Type A) / OK (Type B, Type C)	(Type A reception mode) Error interrupt INTRERR is generated at the Δ point. Measuring the guide pulse is started at the next rising edge. (Type B, Type C reception modes) Measuring the end width is started from the Δ point.

Note In type C reception mode, before the first INTDFULLn interrupt is generated, INTRERRn will not be generated. However, RMSRn and RMSCRn will be cleared.

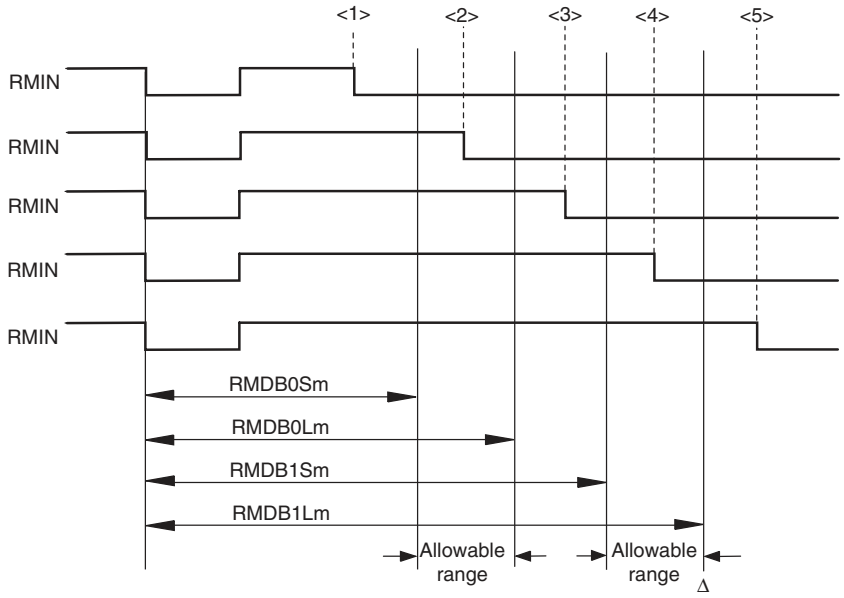
Remark m = 02, 13, n = 0 to 3

(5) Guide pulse cycle determination (Type B1 reception modes only)



Condition		Determination	Subsequent operation
<1>	Counter < RMGPBSm	NG (Short)	Detecting the guide pulse continues and operation restarts from measurement the period of the guide pulse. Measuring guide pulse period is started from this falling edge.
<2>	RMGPBSm ≤ counter < RMGPBLm	OK (Within the range)	INTGPn is generated. Data measurement is started.
<3>	RMGPBLm ≤ counter	NG (Long)	Detecting the guide pulse continues and operation restarts from measurement the period of the guide pulse. Measuring guide pulse period is started from this falling edge.

(6) Data cycle determination (Type B1 and Type C1 reception modes only)



Remark m = 02, 13, n = 0 to 3

Condition		Determination	Subsequent operation
<1>	Counter < RMDB0Sm	NG (Short)	Error interrupt INTRERRn is generated[Notes 1, 2]. (Type B1 reception mode) • RMEROm = 1 Detecting the guide pulse starts. Error information is written to the RMERBSRn register and measuring data period is started from this falling edge. • RMEROm = 0 Detecting the guide pulse starts. Measuring guide pulse period is started from this falling edge. (Type C1 reception mode) • RMEROm = 1 Error information is written to the RMERBSRn register. And then detecting the data continues and operation restarts from measurement the period of the data. Measuring data period is started from this falling edge. • RMEROm = 0 Detecting the data continues and operation restarts from measurement the period of the data. Measuring data period is started from this falling edge.
<2>	RMDB0Sm ≤ counter < RMDB0Lm	OK (Within the range)	Data 0 is received. Detecting the data continues. Measuring data low-level width is started from this falling edge.
<3>	RMDB0Lm ≤ counter < RMDB1Sm	NG (Outside of the range)	Error interrupt INTRERRn is generated[Notes 1, 2]. (Type B1 reception mode) • RMEROm = 1 Detecting the guide pulse starts. Error information is written to the RMERBSRn register and measuring data period is started from this falling edge. • RMEROm = 0 Detecting the guide pulse starts. Measuring guide pulse period is started from this falling edge. (Type C1 reception mode) • RMEROm = 1 Error information is written to the RMERBSRn register. And then detecting the data continues and operation restarts from measurement the period of the data. Measuring data period is started from this falling edge. • RMEROm = 0 Detecting the data continues and operation restarts from measurement the period of the data. Measuring data period is started from this falling edge.
<4>	RMDB1Sm ≤ counter < RMDB1Lm	OK (Within the range)	Data 1 is received. Detecting the data continues. Measuring data low-level width is started from this falling edge.
<5>	RMDB1Lm ≤ counter	OK	Judging the end width is started from the Δ point.

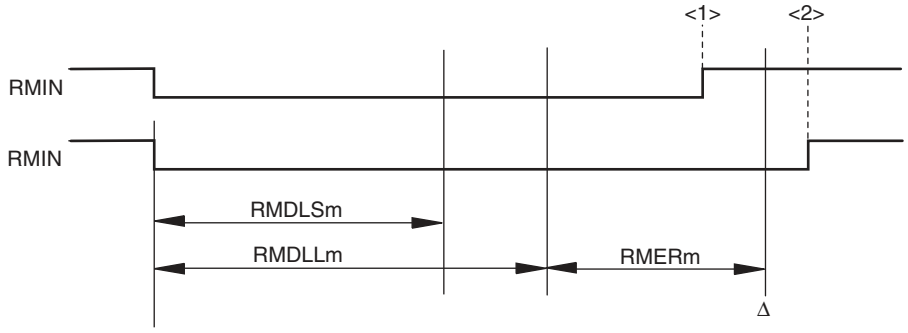
Notes 1. In type C1 reception mode, before the first INTDFULLn interrupt is generated, INTRERRn will not be generated.

2. When an error is generated, processing of RMSRn and RMSCRn depend on RMEROm bit.
- If RMEROm = 0, RMSRn and RMSCRn are cleared at the error generation.
- If RMEROm = 1, RMSRn and RMSCRn are not cleared at the error generation.

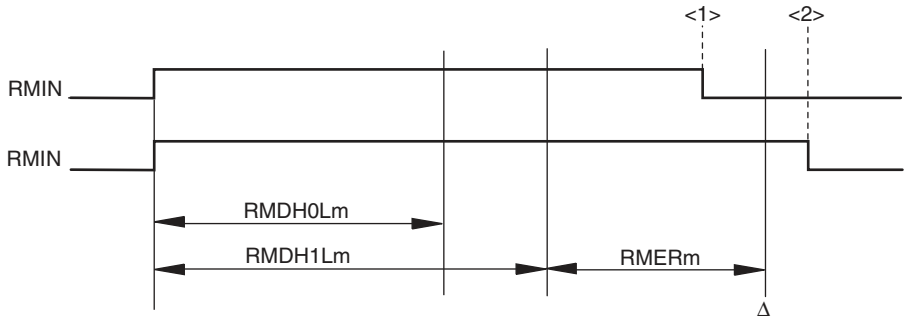
Remark m = 02, 13, n = 0 to 3

(7) End width determination

(a) Type A reception mode



(b) Type B, Type C reception modes

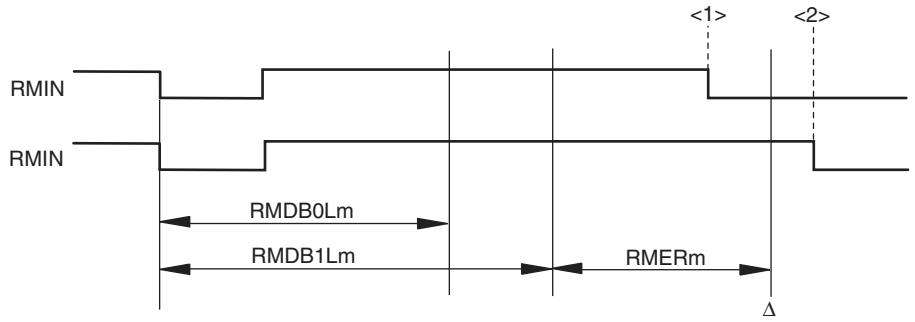


Condition		Determination	Subsequent operation
<1>	Counter < RMERm	NG (Short)	Error interrupt INTRERRn is generated [Note]. (Type A reception mode) Detecting the guide pulse starts. Measuring guide pulse high-level width is started from this rising edge. (Type B reception mode) Detecting the guide pulse starts and operation restarts from determining the low-level width of the guide pulse. Measuring guide pulse low-level width is started from the this falling edge. (Type C reception mode) Detecting the data continues and operation restarts from determining the data low-level width. Measuring data low-level width is started from this falling edge.
<2>	RMERm ≤ counter	OK (Long)	INTRENDn is generated at the Δ point [Note]. Reception stops until RMSRn is read.

Note In type C reception mode, before the first INTDFULLn interrupt is generated, INTRERRn will not be generated. However, RMSRn and RMSCRn will be cleared.

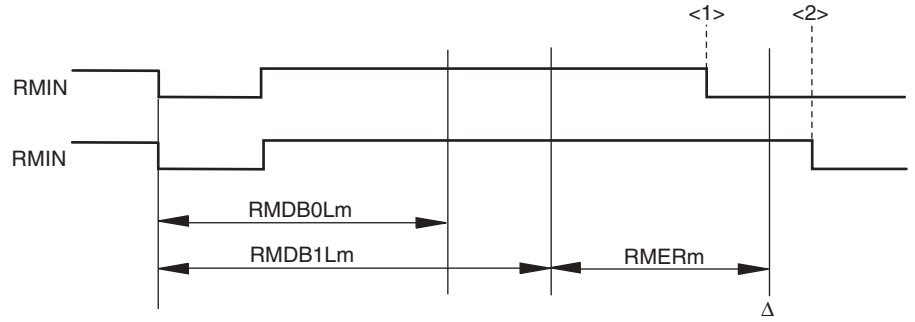
Remark m = 02, 13, n = 0 to 3

(c) Type B1 reception mode



Condition		Determination	Subsequent operation
<1>	Counter < RMERm	NG (Short)	Error interrupt INTRERRn is generated[Note 1]. Detecting the guide pulse starts and operation restarts from measurement the period of the guide pulse. Measuring guide pulse period is started from this falling edge.
<2>	RMERm ≤ counter	OK (Long)	INTRENDn is generated at the Δ point. Reception stops until RMSRn is read.

(d) Type C1 reception mode



Condition		Determination	Subsequent operation
<1>	Counter < RMERm	NG (Short)	Error interrupt INTRERRn is generated[Notes 1, 2]. Detecting the data continues and operation restarts from measurement the period of the data. Measuring data period is started from this falling edge.
<2>	RMERm ≤ counter	OK (Long)	INTRENDn is generated at the Δ point. Reception stops until RMSRn is read.

Notes 1. When an error is generated, processing of RMSRn and RMSCRn depend on RMEROm bit.
- If RMEROm = 0, RMSRn and RMSCRn are cleared at the error generation.
- If RMEROm = 1, RMSRn and RMSCRn are not cleared at the error generation.

2. In type C1 reception mode, before the first INTDFULLn interrupt is generated, INTRERRn will not be generated.

Remark m = 02, 13, n = 0 to 3

14.5.12 Compare register setting

This remote controller receiver has the following compare registers.

- Remote controller receive GPLS compare register m (RMGPLSm)
- Remote controller receive GPLL compare register m (RMGPLLm)
- Remote controller receive GPHS compare register m (RMGPHSm)
- Remote controller receive GPHL compare register m (RMGPHLm)
- Remote controller receive DLS compare register m (RMDLSm)
- Remote controller receive DLL compare register m (RMDLLm)
- Remote controller receive DH0S compare register m (RMDH0Sm)
- Remote controller receive DH0L compare register m (RMDH0Lm)
- Remote controller receive DH1S compare register m (RMDH1Sm)
- Remote controller receive DH1L compare register m (RMDH1Lm)
- Remote controller receive GPBS compare register m (RMGPBSm)
- Remote controller receive GPBL compare register m (RMGPBLm)
- Remote controller receive DB0S compare register m (RMDB0Sm)
- Remote controller receive DB0L compare register m (RMDB0Lm)
- Remote controller receive DB1S compare register m (RMDB1Sm)
- Remote controller receive DB1L compare register m (RMDB1Lm)
- Remote controller receive end width select register m (RMERHm/RMERLm)
- Remote controller receive noise elimination period setting register m (RMNCP1m)

Use formulas (1) to (3) below to set the value of each compare register.

Making allowances for tolerance enables a normal reception operation, even if the RMIN input waveform is RMIN_1 or RMIN_2 shown in Figures 14-42 and 14-43 due to the effect of noise.

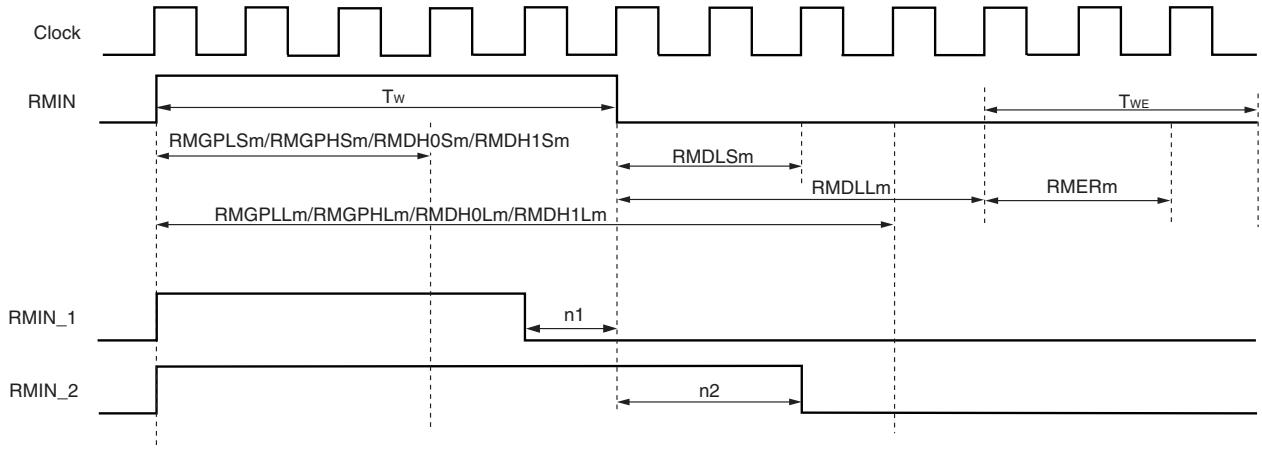
Cautions 1. Always set each compare register while remote controller reception is disabled (RMENm = 0).

2. Set the set values so that they satisfy all the following four conditions.

- **RMGPLSm < RMGPLLm**
- **RMGPHSm < RMGPHLm**
- **RMDLSm < RMDLLm**
- **RMDH0Sm < RMDH0Lm ≤ RMDH1Sm < RMDH1Lm**
- **RMGPBSm < RMGPBLm**
- **RMDB0Sm < RMDB0Lm ≤ RMDB1Sm < RMDB1Lm**
- **RMNCP1m < RMGPBSm, RMGPBLm, RMDB0Sm, RMDB0Lm, RMDB1Sm, RMDB1Lm, RMERm**

Remark m = 02, 13

Figure 14-42. Setting Example in Type A Reception Mode (Where n1 = 1, n2 = 2)



(1) Formula for RMGPLSm, RMGPHSm, RMDLSm, RMDH0Sm, and RMDH1Sm

$$\left[\frac{T_W \times (1 - a/100)}{1/f_{REMPRS}}\right]_{INT} - 2 - n1$$

(2) Formula for RMGPLLm, RMGPHLm, RMDLLm, RMDH0Lm, and RMDH1Lm

$$\left[\frac{T_W \times (1 + a/100)}{1/f_{REMPRS}}\right]_{INT} + 1 + n2$$

(3) Formula for RMERm

$$\left[\frac{T_{WE} \times (1 - a/100)}{1/f_{REMPRS}}\right]_{INT} - 1$$

T_W: Width of RMIN input waveform

$1/f_{REMPRS}$: Width of internal operation clock cycle after division control by RMPRSm

a: Tolerance (%)

[ ] INT: Round down the fractional portion of the value produced by the formula in the brackets.

n1, n2: Variables of waveform change caused by noise[Note 1]

T_{WE}: End width of RMIN input[Note 2]

Notes 1. Set the values of n1 and n2 as required to meet the user's system specification.

2. This end width is counted after RMDLLm.

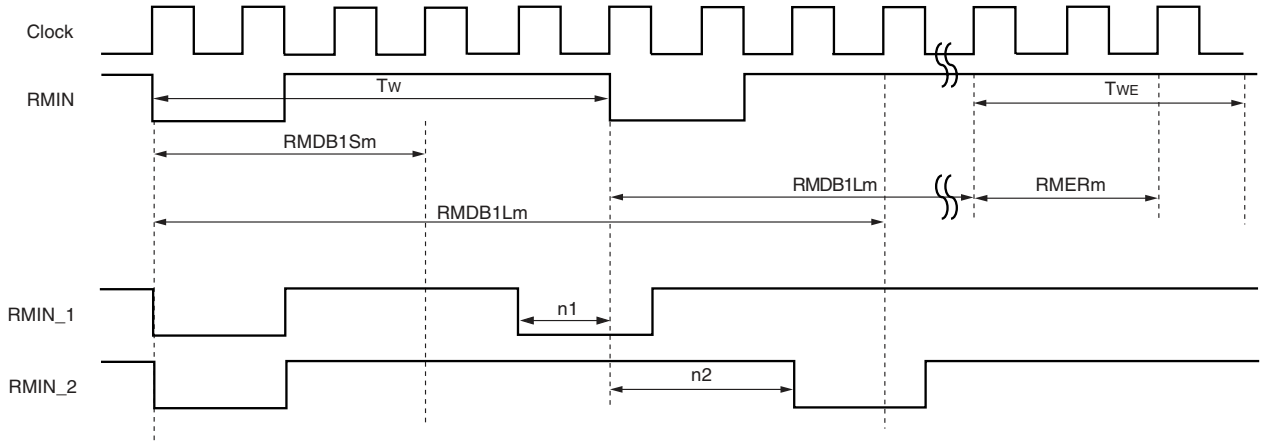
The low-level width actually required after the last data has been received is as follows:

(RMDLLm + 1 + RMERm + 1)× (width of internal operation clock cycle after division control by RMPRSm)

If this cannot be met, INTRERRn will be generated.

Remark m = 02, 13, n = 0 to 3

Figure 14-43. Setting Example in Type B1 and Type C1 Reception Mode (Where n1 = 1, n2 = 2)



(1) Formula for RMDB1Sm

$$\left(\frac{Tw \times (1 - a/100)}{1/f_{REMPRS}} \right)_{INT} - 2 - n1$$

(2) Formula for RMDB1Lm

$$\left(\frac{Tw \times (1 + a/100)}{1/f_{REMPRS}} \right)_{INT} + 1 + n2$$

(3) Formula for RMERm

$$\left(\frac{TwE \times (1 - a/100)}{1/f_{REMPRS}} \right)_{INT} - 1$$

Tw: Width of RMIN input waveform

$1/f_{REMPRS}$: Width of internal operation clock cycle after division control by RMPRSm

a: Tolerance (%)

[ ] INT: Round down the fractional portion of the value produced by the formula in the brackets.

n1, n2: Variables of waveform change caused by noise[Note 1]

TwE: End width of RMIN input[Note 2]

Notes 1. Set the values of n1 and n2 as required to meet the user's system specification.

2. This end width is counted after RMDB1Lm.

The last data cycle width actually required is as follows:

(RMDB1Lm + 1 + RMERm + 1)× (width of internal operation clock cycle after division control by RMPRSm)

If this cannot be met, INTRERRn is generated.

Remark m = 02, 13, n = 0 to 3

14.5.13 Error interrupt generation timing

(1) Type A reception mode

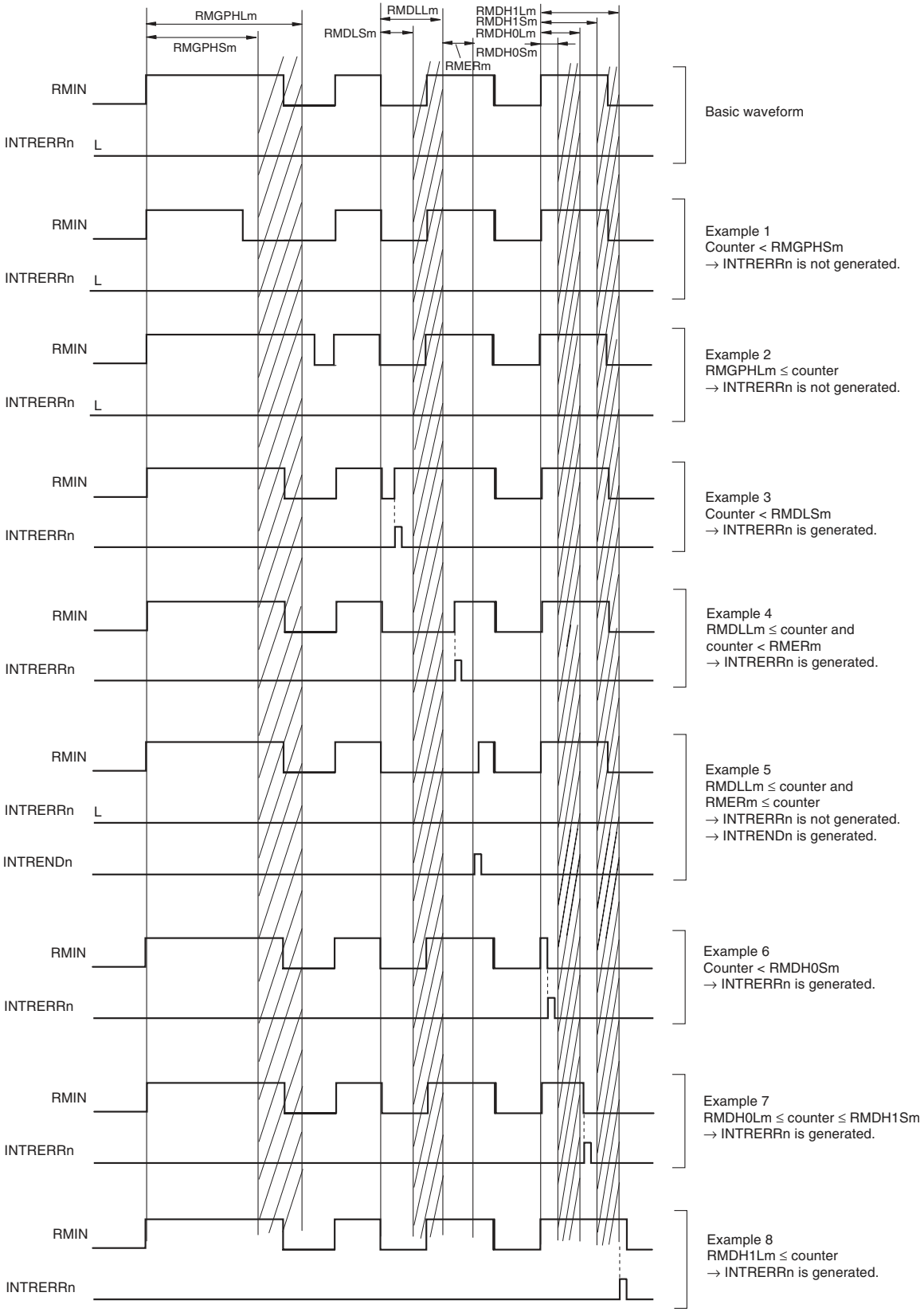
After the guide pulse has been detected normally, the INTRERRn signal is generated under any of the following conditions.

- Counter < RMDLSm at the rising edge of RMIN
- RMDLLm ≤ counter and counter after RMDLLm < RMERm at the rising edge of RMIN
- Counter < RMDH0Sm at the falling edge of RMIN
- RMDH0Lm ≤ counter < RMDH1Sm at the falling edge of RMIN
- RMDH1Lm ≤ counter while RMIN is at high level

The INTRERRn signal is not generated until the guide pulse is detected.
Once the INTRERRn signal has been generated, it will not be generated again until the next guide pulse is detected. The generation timing of the INTRERRn signal is shown in **Figure 14-44**.

Remark m = 02, 13, n = 0 to 3

Figure 14-44. Generation Timing of INTRERRn Signal (Type A reception mode)



Remark m = 02, 13, n = 0 to 3

(2) Type B reception mode

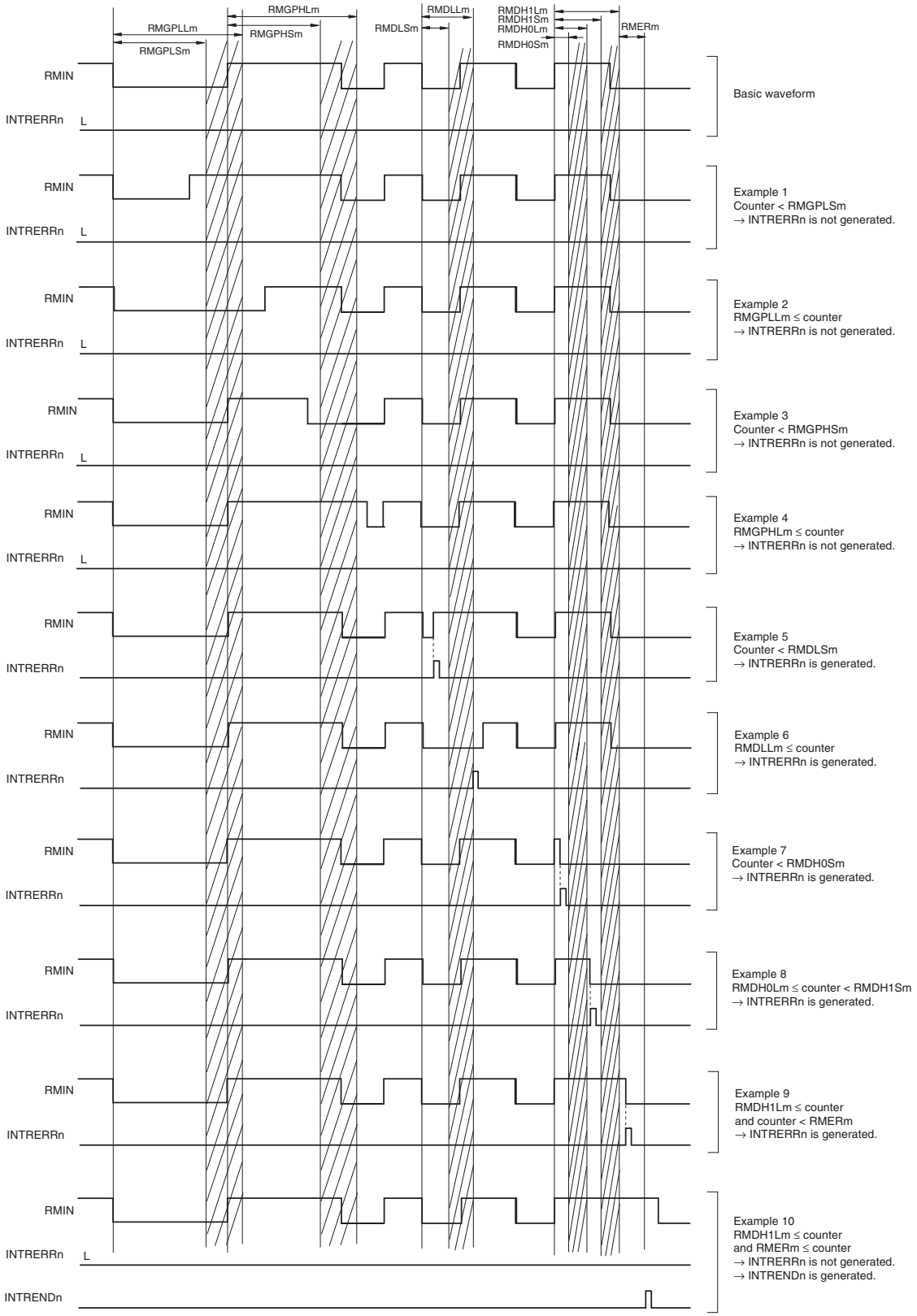
After the guide pulse has been detected normally, the INTRERRn signal is generated under any of the following conditions.

- Counter < RMDLSm at the rising edge of RMIN
- RMDLLm ≤ counter while RMIN is at low level
- Counter < RMDH0Sm at the falling edge of RMIN
- RMDH0Lm ≤ counter < RMDH1Sm at the falling edge of RMIN
- RMDH1Lm ≤ counter and counter after RMDH1Lm < RMERm at the falling edge of RMIN

The INTRERRn signal is not generated until the guide pulse is detected.
Once the INTRERRn signal has been generated, it will not be generated again until the next guide pulse is detected. The generation timing of the INTRERRn signal is shown in **Figure 14-45**.

Remark m = 02, 13, n = 0 to 3

Figure 14-45. Generation Timing of INTRERRn Signal (Type B reception mode)



Remark m = 02, 13, n = 0 to 3

(3) Type C reception mode

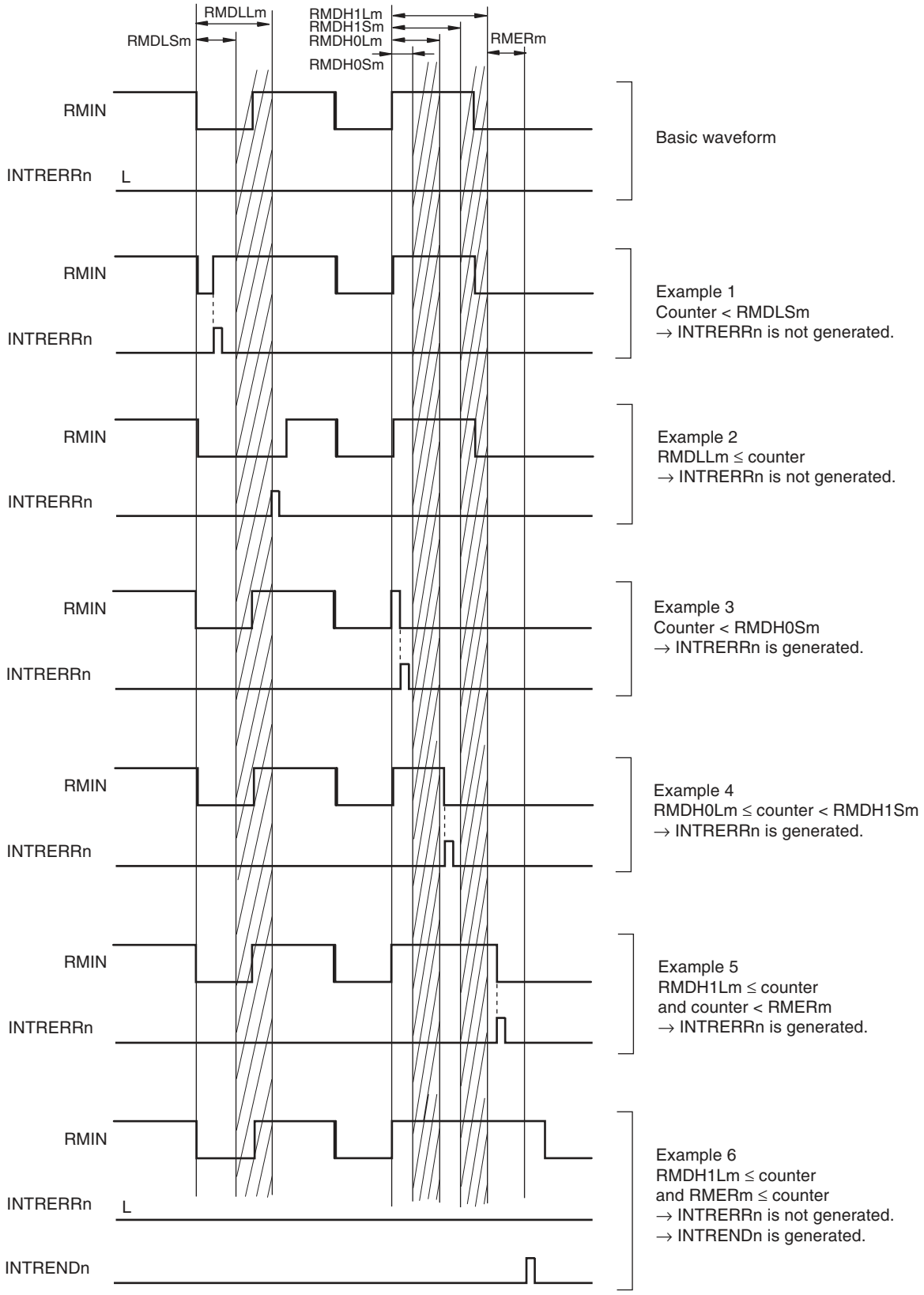
The INTRERRn signal is generated under any of the following conditions.

- Counter < RMDLSm at the rising edge of RMIN
- RMDLLm ≤ counter while RMIN is at low level
- Counter < RMDH0Sm at the falling edge of RMIN
- RMDH0Lm ≤ counter < RMDH1Sm at the falling edge of RMIN
- RMDH1Lm ≤ counter and counter after RMDH1Lm < RMERm at the falling edge of RMIN

However, before the first INTDFULLn interrupt is generated, INTRERRn signal will not be generated.
The generation timing of the INTRERRn signal is shown in **Figure 14-46**.

Remark m = 02, 13, n = 0 to 3

Figure 14-46. Generation Timing of INTRERRn Signal (Type C reception mode)



Remark m = 02, 13, n = 0 to 3

14.5.14 Instruction for interrupt use

The following three interrupt sources are output as INTRAn signals in this remote controller receiver.

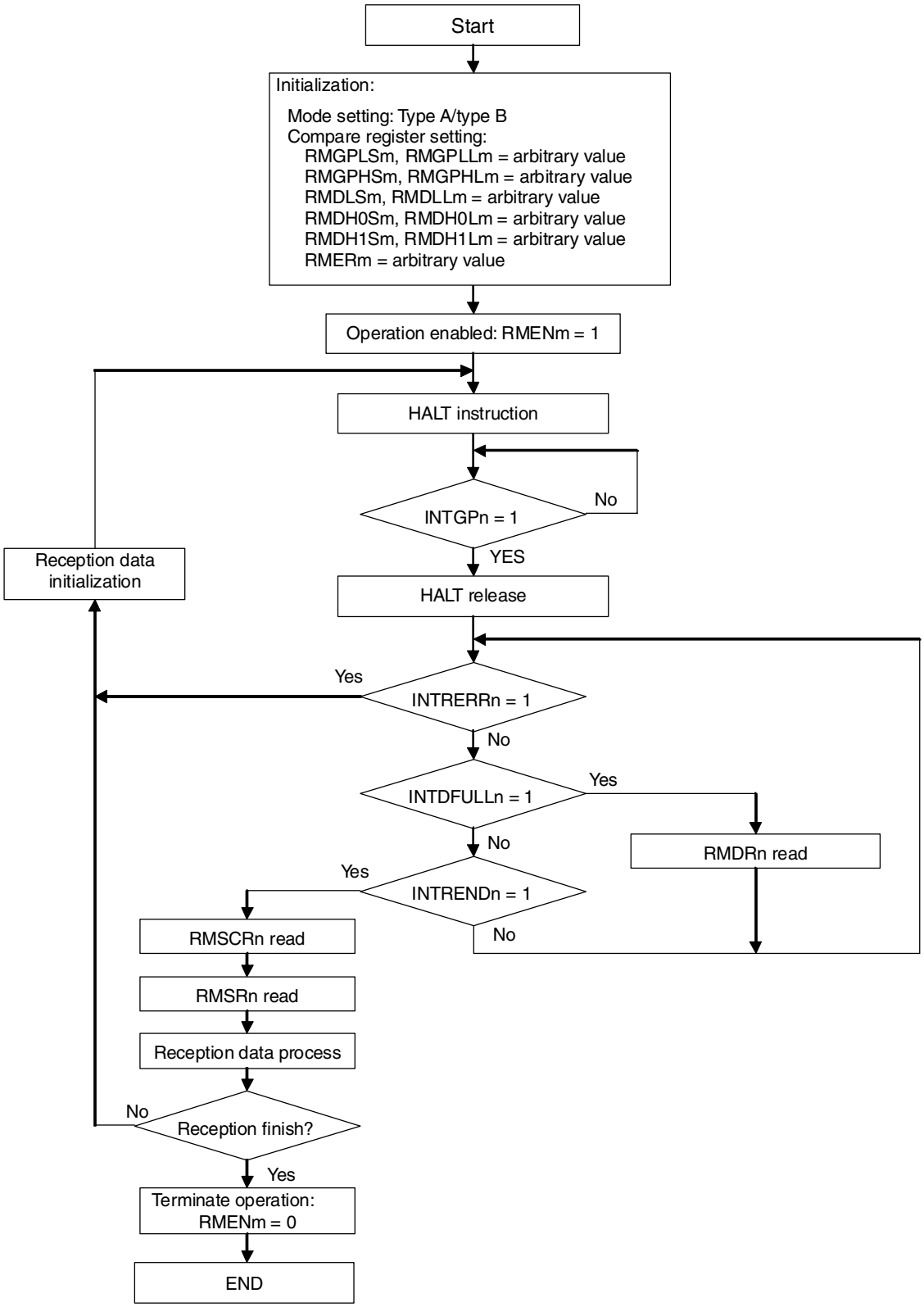
- 8-bit data reception completion (INTDFULLn)
- Reception completion (INTRENDn)
- Guide pulse detection (INTGPn)

In this chapter, the interrupt sources are expressed as INTDFULLn, INTRENDn, and INTGPn for the sake of explanation, but should be read as follows.

- INTDFULLn: INTRAn interrupt occurrence and INTSDFULLn of the RMINTSn register = 1
- INTRENDn: INTRAn interrupt occurrence and INTSRENDn of the RMINTSn register = 1
- INTGPn: INTRAn interrupt occurrence and INTSGPn of the RMINTSn register = 1

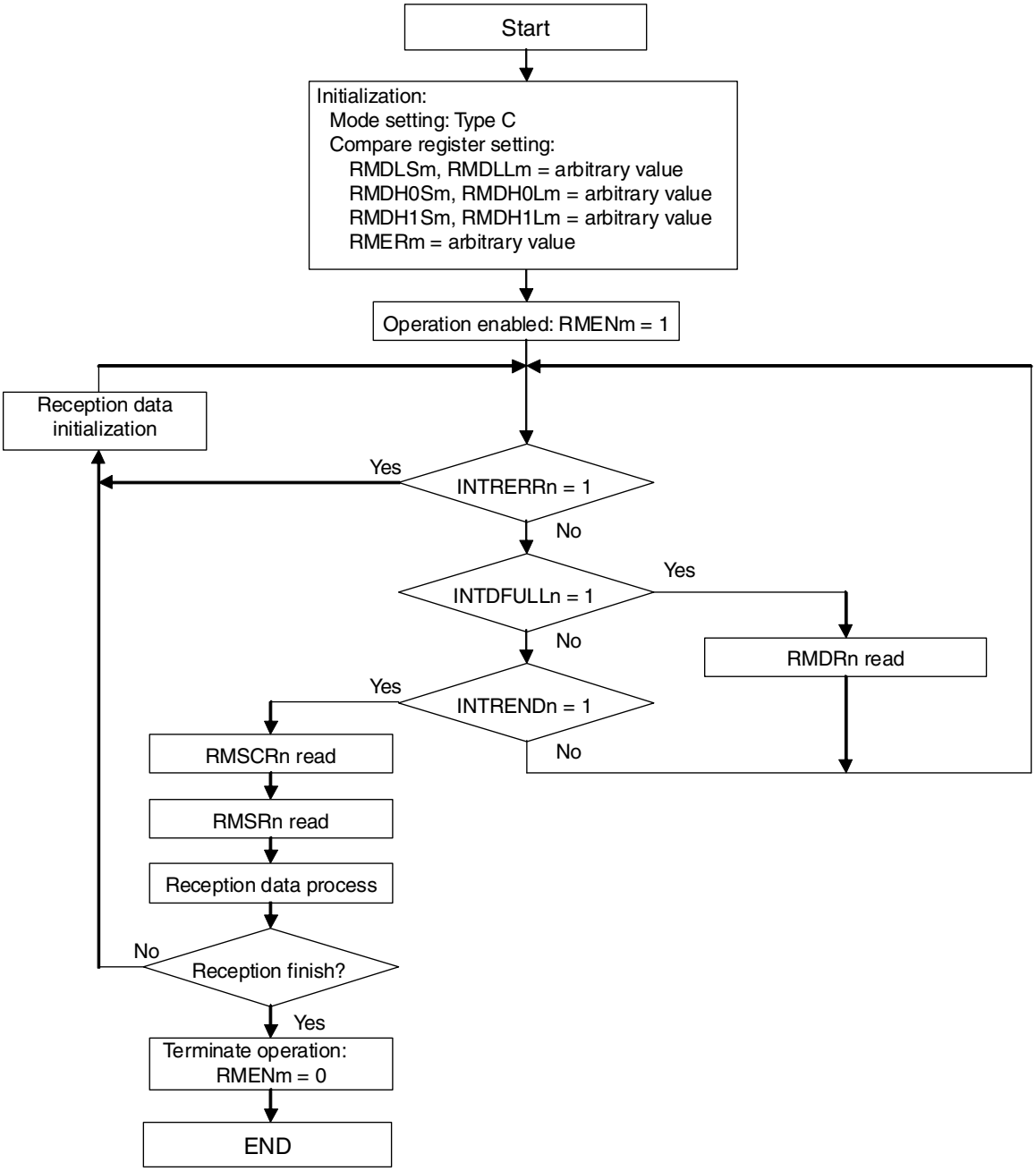
Remark n = 0 to 3

Figure 14-47. S/W Flow of INTGPn, INTRERRn, INTRENDn, INTDFULLn Use (Type A/Type B Reception Mode)



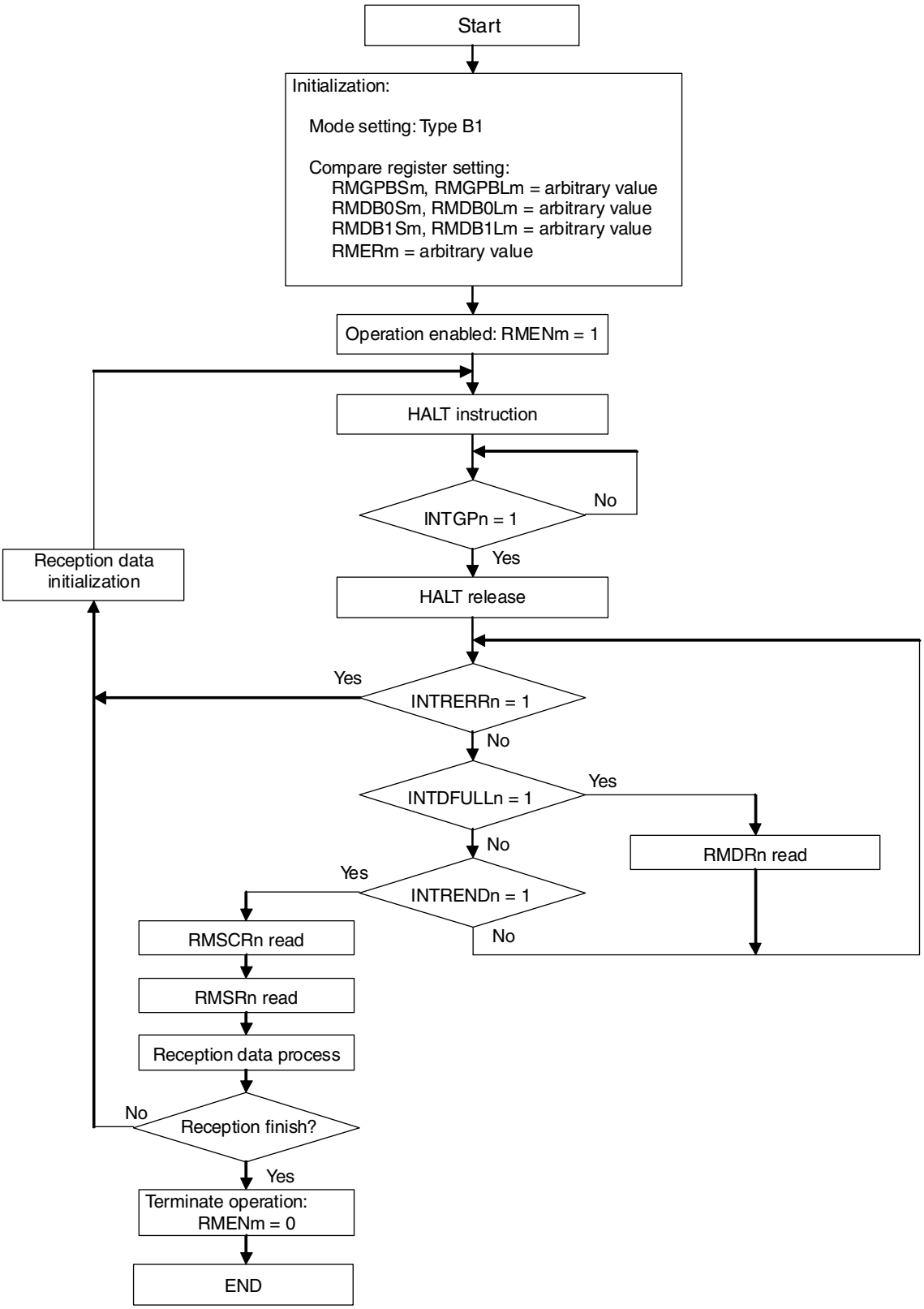
Remark m = 02, 13, n = 0 to 3

Figure 14-48. S/W Flow of INTRERRn, INTRENDn, INTDFULLn Use (Type C Reception Mode)



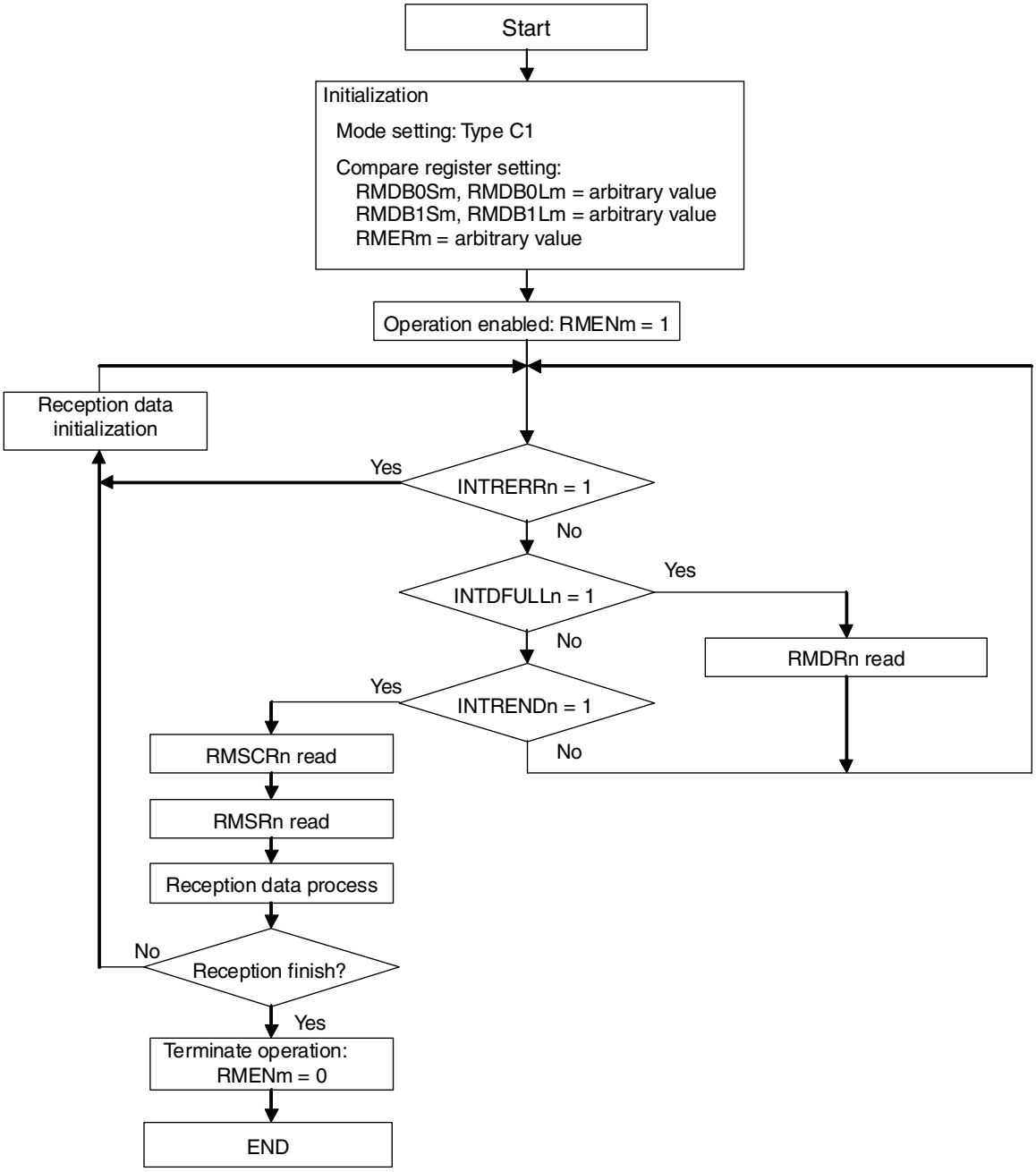
Remark m = 02, 13, n = 0 to 3

Figure 14-49. S/W Flow of INTGPn, INTRERRn, INTRENDn, INTDFULLn Use (Type B1 Reception Mode)



Remark m = 02, 13, n = 0 to 3

Figure 14-50. S/W Flow of INTRERRn, INTRENDn, INTDFULLn Use (Type C1 Reception Mode)



Remark m = 02, 13, n = 0 to 3

CHAPTER 15 MULTIPLIER/DIVIDER

15.1 Functions of Multiplier/Divider

The multiplier/divider has the following functions.

- 16 bits × 16 bits = 32 bits (multiplication)
- 32 bits ÷ 32 bits = 32 bits, 32-bit remainder (division)

15.2 Configuration of Multiplier/Divider

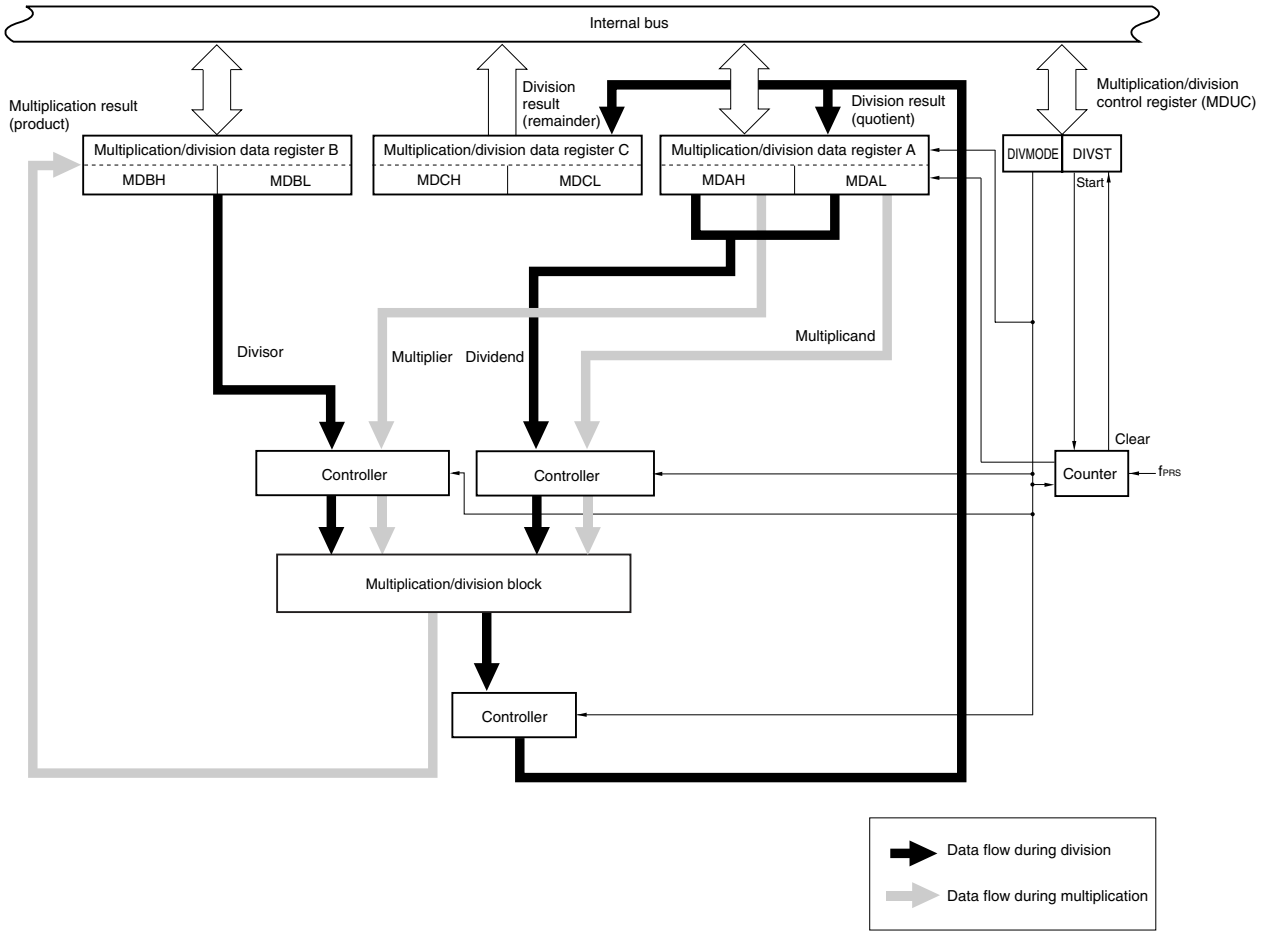
The multiplier/divider consists of the following hardware.

Table 15-1. Configuration of Multiplier/Divider

Item	Configuration
Registers	Multiplication/division data register A (L) (MDAL) Multiplication/division data register A (H) (MDAH) Multiplication/division data register B (L) (MDBL) Multiplication/division data register B (H) (MDBH) Multiplication/division data register C (L) (MDCL) Multiplication/division data register C (H) (MDCH)
Control register	Multiplication/division control register (MDUC)

Figure 15-1 shows a block diagram of the multiplier/divider.

Figure 15-1. Block Diagram of Multiplier/Divider



(1) Multiplication/division data register A (MDAH, MDAL)

The MDAH and MDAL registers set the values that are used for a multiplication or division operation and store the operation result. They set the multiplier and multiplicand data in the multiplication mode, and set the dividend data in the division mode. Furthermore, the operation result (quotient) is stored in the MDAH and MDAL registers in the division mode.

MDAH and MDAL can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 15-2. Format of Multiplication/Division Data Register A (MDAH, MDAL)

Address: FFFF0H, FFFF1H, FFFF2H, FFFF3H After reset: 0000H, 0000H R/W

Symbol	FFFF3H								FFFF2H							
MDAH	MDAH15	MDAH14	MDAH13	MDAH12	MDAH11	MDAH10	MDAH9	MDAH8	MDAH7	MDAH6	MDAH5	MDAH4	MDAH3	MDAH2	MDAH1	MDAH0

Symbol	FFFF1H								FFFF0H							
MDAL	MDAL15	MDAL14	MDAL13	MDAL12	MDAL11	MDAL10	MDAL9	MDAL8	MDAL7	MDAL6	MDAL5	MDAL4	MDAL3	MDAL2	MDAL1	MDAL0

Cautions 1. Do not rewrite the MDAH and MDAL values during division operation processing (while the multiplication/division control register (MDUC) is 81H). The operation will be executed in this case, but the operation result will be an undefined value.

2. The MDAH and MDAL values read during division operation processing (while MDUC is 81H) will not be guaranteed.

The following table shows the functions of MDAH and MDAL during operation execution.

Table 15-2. Functions of MDAH and MDAL During Operation Execution

DIVMODE	Operation Mode	Setting	Operation Result
0	Multiplication mode	MDAH: Multiplier MDAL: Multiplicand	–
1	Division mode	MDAH: Divisor (higher 16 bits) MDAL: Dividend (lower 16 bits)	MDAH: Division result (quotient) Higher 16 bits MDAL: Division result (quotient) Lower 16 bits

Remark DIVMODE: Bit 7 of the multiplication/division control register (MDUC)

(2) Multiplication/division data register B (MDBL, MDBH)

The MDBH and MDBL registers set the values that are used for multiplication or division operation and store the operation result. They store the operation result (product) in the multiplication mode and set the divisor data in the division mode.

MDBH and MDBL can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 15-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)

Address: FFFF4H, FFFF5H, FFFF6H, FFFF7H After reset: 0000H, 0000H R/W

Symbol	FFFF5H								FFFF4H							
MDBH	MDBH 15	MDBH 14	MDBH 13	MDBH 12	MDBH 11	MDBH 10	MDBH 9	MDBH 8	MDBH 7	MDBH 6	MDBH 5	MDBH 4	MDBH 3	MDBH 2	MDBH 1	MDBH 0

Symbol	FFFF7H								FFFF6H							
MDBL	MDBL 15	MDBL 14	MDBL 13	MDBL 12	MDBL 11	MDBL 10	MDBL 9	MDBL 8	MDBL 7	MDBL 6	MDBL 5	MDBL 4	MDBHL 3	MDBL 2	MDBL 1	MDBL 0

Cautions 1. Do not rewrite the MDBH and MDBL values during division operation processing (while the multiplication/division control register (MDUC) is 81H). The operation result will be an undefined value.

2. Do not set MDBH and MDBL to 0000H in the division mode. If they are set, the operation result will be an undefined value.

The following table shows the functions of MDBH and MDBL during operation execution.

Table 15-3. Functions of MDBH and MDBL During Operation Execution

DIVMODE	Operation Mode	Setting	Operation Result
0	Multiplication mode	–	MDBH: Multiplication result (product) Higher 16 bits MDBL: Multiplication result (product) Lower 16 bits
1	Division mode	MDBH: Divisor (higher 16 bits) MDBL: Dividend (lower 16 bits)	–

Remark DIVMODE: Bit 7 of the multiplication/division control register (MDUC)

(3) Multiplication/division data register C (MDCL, MDCH)

The MDCH and MDCL registers store remainder value of the operation result in the division mode. They are not used in the multiplication mode.

MDCH and MDCL can be read by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 15-4. Format of Multiplication/Division Data Register C (MDCH, MDCL)

Address: F00E0H, F00E1H, F00E2H, F00E3H After reset: 0000H, 0000H R

Symbol F00E3H F00E2H

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

Symbol F00E1H F00E0H

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

Caution The MDCH and MDCL values read during division operation processing (while the multiplication/division control register (MDUC) is 81H) will not be guaranteed.

Table 15-4. Functions of MDCH and MDCL During Operation Execution

DIVMODE	Operation Mode	Setting	Operation Result
0	Multiplication mode	–	–
1	Division mode	–	MDCH: Remainder (higher 16 bits) MDCL: Remainder (lower 16 bits)

Remark DIVMODE: Bit 7 of the multiplication/division control register (MDUC)

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

<Multiplier A> <Multiplier B> <Product>

MDAL (bits 15 to 0) × MDAH (bits 15 to 0) = [MDBH (bits 15 to 0), MDBL (bits 15 to 0)]

- Register configuration during division

<Dividend> <Divisor>

[MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ÷ [MDBH (bits 15 to 0), MDBL (bits 15 to 0)] =

<Quotient> <Remainder>

[MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ··· [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

15.3 Register Controlling Multiplier/Divider

The multiplier/divider is controlled by using the multiplication/division control register (MDUC).

(1) Multiplication/division control register (MDUC)

MDUC is an 8-bit register that controls the operation of the multiplier/divider.

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

Reset signal generation clears this register to 00H.

Figure 15-5. Format of Multiplication/Division Control Register (MDUC)

Address: F00E8H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
MDUC	DIVMODE	0	0	0	0	0	0	DIVST

DIVMODE	Operation mode (multiplication/division) selection
0	Multiplication mode
1	Division mode

DIVST[Note]	Division operation start/stop
0	Division operation processing complete
1	Starts division operation/division operation processing in progress

Note DIVST can only be set (1) in the division mode. In the division mode, division operation is started by setting (1) DIVST. DIVST is automatically cleared (0) when the operation ends. In the multiplication mode, operation is automatically started by setting the multiplier and multiplicand to MDAH and MDAL, respectively.

Cautions 1. Do not rewrite DIVMODE during operation processing (while DIVST is 1). If it is rewritten, the operation result will be an undefined value.

2. DIVST cannot be cleared (0) by using software during division operation processing (while DIVST is 1).

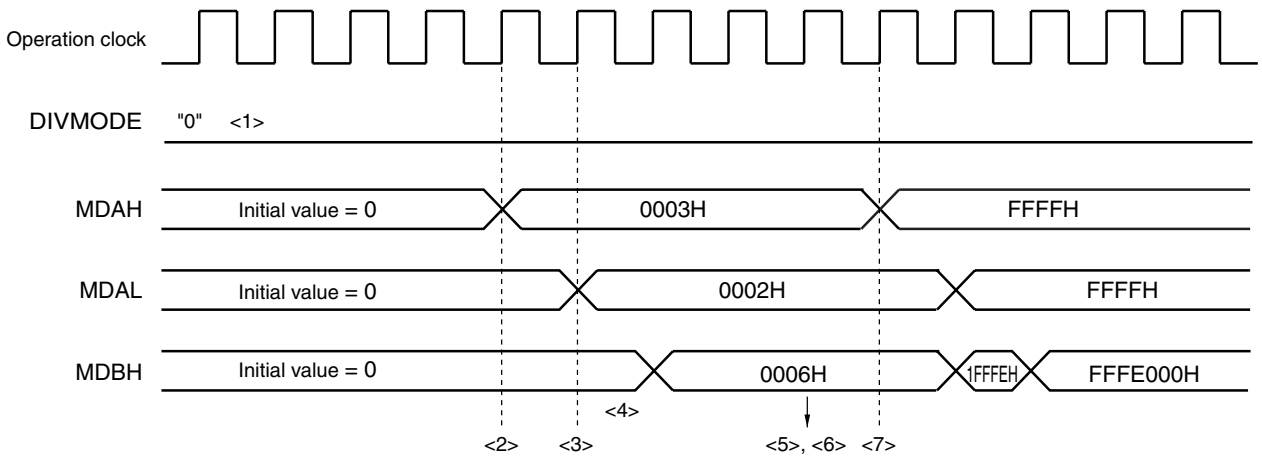
15.4 Operations of Multiplier/Divider

15.4.1 Multiplication operation

- Initial setting
  - <1> Set bit 7 (DIVMODE) of the multiplication/division control register (MDUC) to 0.
  - <2> Set the multiplicand to the multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to the multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>. Multiplication operation is automatically started when the multiplier and multiplicand are set to MDAH and MDAL, respectively.)
- During operation processing
  - <4> Wait for at least one clock. The operation will end when one clock has been issued.
- Operation end
  - <5> Read the product (lower 16 bits) from the multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from the multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation
  - <7> To execute multiplication operation next, start from the “Initial setting” for multiplication operation.
  - <8> To execute division operation next, start from the “Initial setting” in **15.4.2 Division operation**.

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 15-6.

Figure 15-6. Timing Diagram of Multiplication Operation (0003H × 0002H)



15.4.2 Division operation

- Initial setting
  - <1> Set bit 7 (DIVMODE) of the multiplication/division control register (MDUC) to 1.
  - <2> Set the dividend (higher 16 bits) to the multiplication/division data register A (H) (MDAH).
  - <3> Set the dividend (lower 16 bits) to the multiplication/division data register A (L) (MDAL).
  - <4> Set the divisor (higher 16 bits) to the multiplication/division data register B (H) (MDBH).
  - <5> Set the divisor (lower 16 bits) to the multiplication/division data register B (L) (MDBL).
  - <6> Set bit 0 (DIVST) of MDUC to 1.

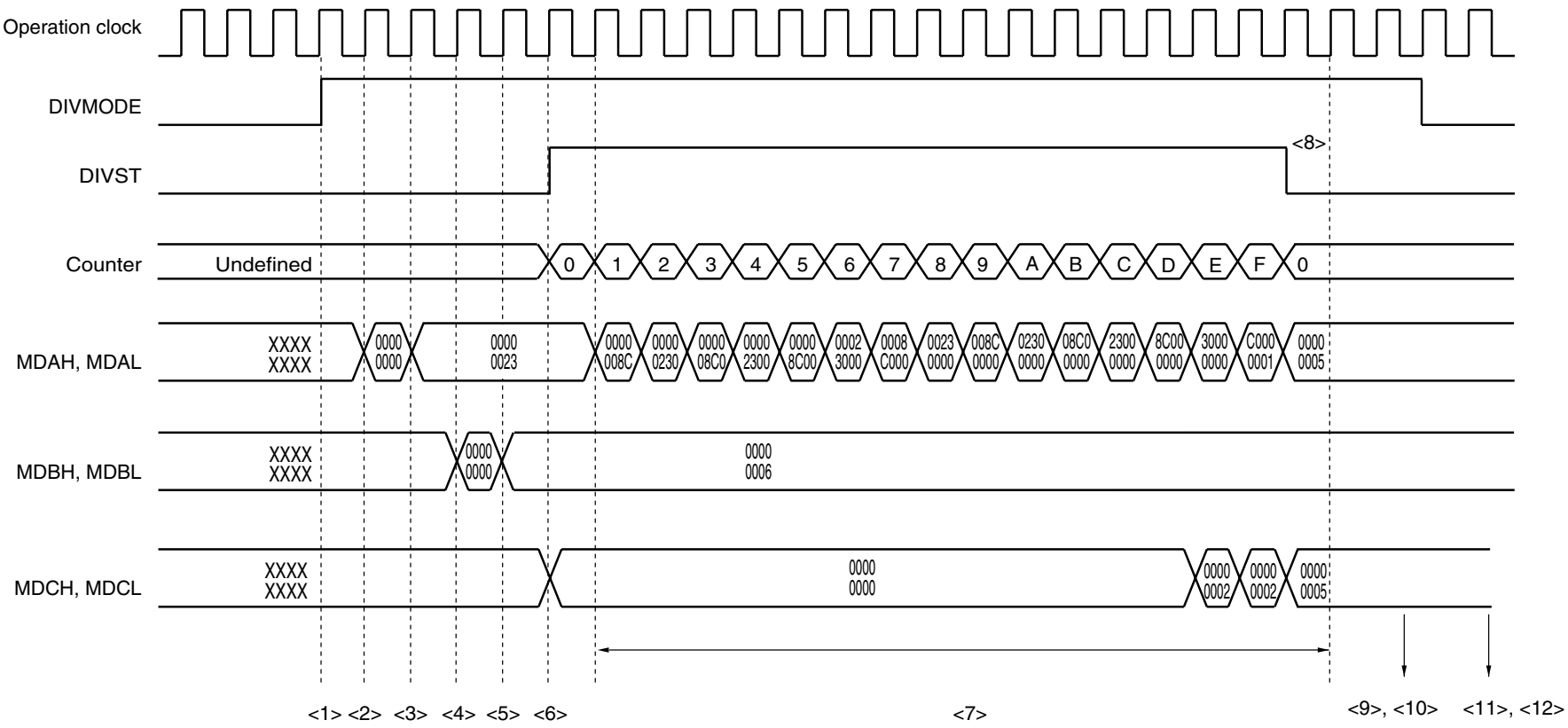
    (There is no preference in the order of executing steps <2> to <5>.)
- During operation processing
  - <7> The operation will end when one of the following processing is completed.
    - A wait of at least 16 clocks (The operation will end when 16 clocks have been issued.)
    - A check whether DIVST has been cleared

    (The read values of MDBL, MDBH, MDCH, and MDCL during operation processing are not guaranteed.)
- Operation end
  - <8> DIVST is cleared (0) (end of operation).
  - <9> Read the quotient (lower 16 bits) from MDAL.
  - <10> Read the quotient (higher 16 bits) from MDAH.
  - <11> Read the remainder (lower 16 bits) from multiplication/division data register C (L) (MDCL).
  - <12> Read the remainder (higher 16 bits) from the multiplication/division data register C (H) (MDCH).

    (There is no preference in the order of executing steps <9> to <12>.)
- Next operation
  - <13> To execute multiplication operation next, start from the “Initial setting” in **15.4.1 Multiplication operation**.
  - <14> To execute division operation next, start from the “Initial setting” for division operation.

Remark Steps <1> to <12> correspond to <1> to <12> in Figure 15-7.

Figure 15-7. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



CHAPTER 16 DMA CONTROLLER

The 78K0R/Kx3-C has an internal DMA (Direct Memory Access) controller.

Data can be automatically transferred between SFRs of the peripheral hardware supporting DMA 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 (CSI00, CSI01, CSI10, UART0, UART1, or IIC10)
  - Timer (channel 0, 1, 4, or 5)
- 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	• DMA SFR address registers 0, 1 (DSA0, DSA1) • DMA RAM address registers 0, 1 (DRA0, DRA1)
Count register	• DMA byte count registers 0, 1 (DBC0, DBC1)
Control registers	• DMA mode control registers 0, 1 (DMC0, DMC1) • DMA operation control registers 0, 1 (DRC0, DRC1)

(1) DMA SFR address register n (DSAn)

This is an 8-bit register that is used to set an SFR address that is the transfer source or destination of DMA channel n.

Set the lower 8 bits of the SFR addresses FFF00H to FFFFFH.

This register is not automatically incremented but fixed to a specific value.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

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.

Figure 16-1. Format of DMA SFR Address Register n (DSAn)

Address: FFFB0H (DSA0), FFFB1H (DSA1) After reset: 00H R/W

	7	6	5	4	3	2	1	0
DSAn								

Remark n: DMA channel number (n = 0, 1)

(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 (FE700H to FFEDFH in the case of the μPD78F1846A, 78F1848A) 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.

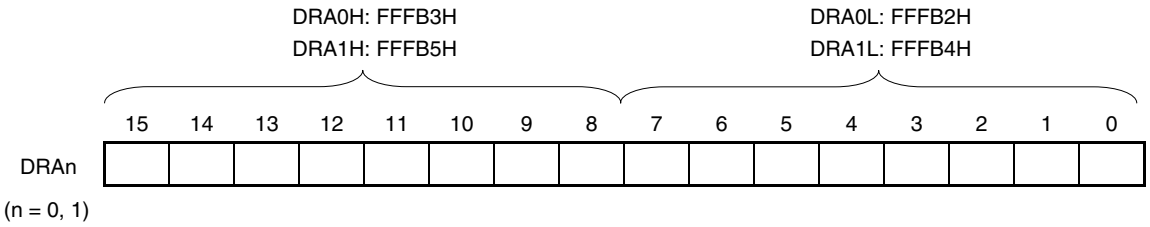
In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

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)

Address: FFFB2H, FFFB3H (DRA0), FFFB4H, FFFB5H (DRA1) After reset: 0000H R/W



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).

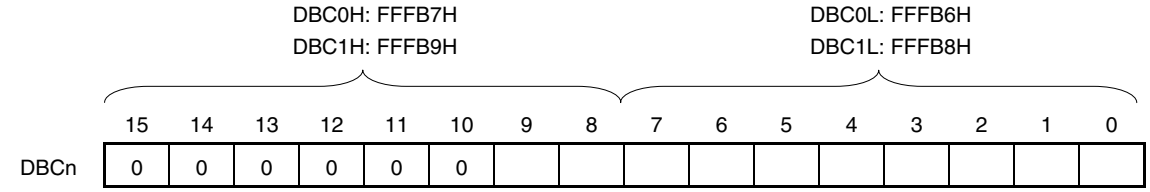
Each time DMA transfer has been executed, this register is automatically decremented. By reading this DBCn register during DMA transfer, the remaining number of times of transfer can be learned.

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)

Address: FFFB6H, FFFB7H (DBC0), FFFB8H, FFFB9H (DBC1) After reset: 0000H R/W



(n = 0, 1)

DBCn[9:0]	Number of Times of Transfer (When DBCn is Written)	Remaining Number of Times of Transfer (When DBCn is Read)
000H	1024	Completion of transfer or waiting for 1024 times of DMA transfer
001H	1	Waiting for remaining one time of DMA transfer
002H	2	Waiting for remaining two times of DMA transfer
003H	3	Waiting for remaining three times of DMA transfer
• • •	• • •	• • •
3FEH	1022	Waiting for remaining 1022 times of DMA transfer
3FFH	1023	Waiting for remaining 1023 times of DMA transfer

Cautions 1. Be sure to clear bits 15 to 10 to "0".

2. If the general-purpose register is specified or the internal RAM space is exceeded as a result of continuous transfer, the general-purpose register or SFR space are written or read, resulting in loss of data in these spaces. Be sure to set the number of times of transfer that is within the internal RAM space.

Remark n: DMA channel number (n = 0, 1)

16.3 Registers Controlling DMA Controller

DMA controller is controlled by the following registers.

- DMA mode control register n (DMCn)
- DMA operation control register n (DRCn)

Remark n: DMA channel number (n = 0, 1)

(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[Note 1]	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).
DMA transfer is started 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[Note 2]	Pending of DMA transfer
0	Executes DMA transfer upon DMA start request (not held pending).
1	Holds DMA start request pending if any.
DMA transfer that has been held pending can be started by clearing the value of 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 the DWAIT0 and DWAIT1 bits to 1).

Remark n: DMA channel number (n = 0, 1)

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 start source[Note]	
				Trigger signal	Trigger contents
0	0	0	0	–	Disables DMA transfer by interrupt. (Only software trigger is enabled.)
0	0	1	0	INTTM00	Timer channel 0 interrupt
0	0	1	1	INTTM01	Timer channel 1 interrupt
0	1	0	0	INTTM04	Timer channel 4 interrupt
0	1	0	1	INTTM05	Timer channel 5 interrupt
0	1	1	0	INTST0/INTCSI00	UART0 transmission end interrupt/CSI00 transfer end interrupt
0	1	1	1	INTSR0/INTCSI01	UART0 reception end interrupt/CSI01 transfer end interrupt
1	0	0	0	INTST1/INTCSI10/INTIIC10	UART1 transmission end interrupt/ CSI10 transfer end interrupt/ IIC10 transfer end interrupt
1	0	0	1	INTSR1	UART1 reception 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.
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).
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.	

Cautions 1. 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 Forced termination by software).

2. When the FSEL bit of the OSMC register has been set to 1, do not enable (DENn = 1) DMA operation for at least three clocks after the setting.

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 DENn = 1. Before writing the other registers, be sure to set DENn 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.

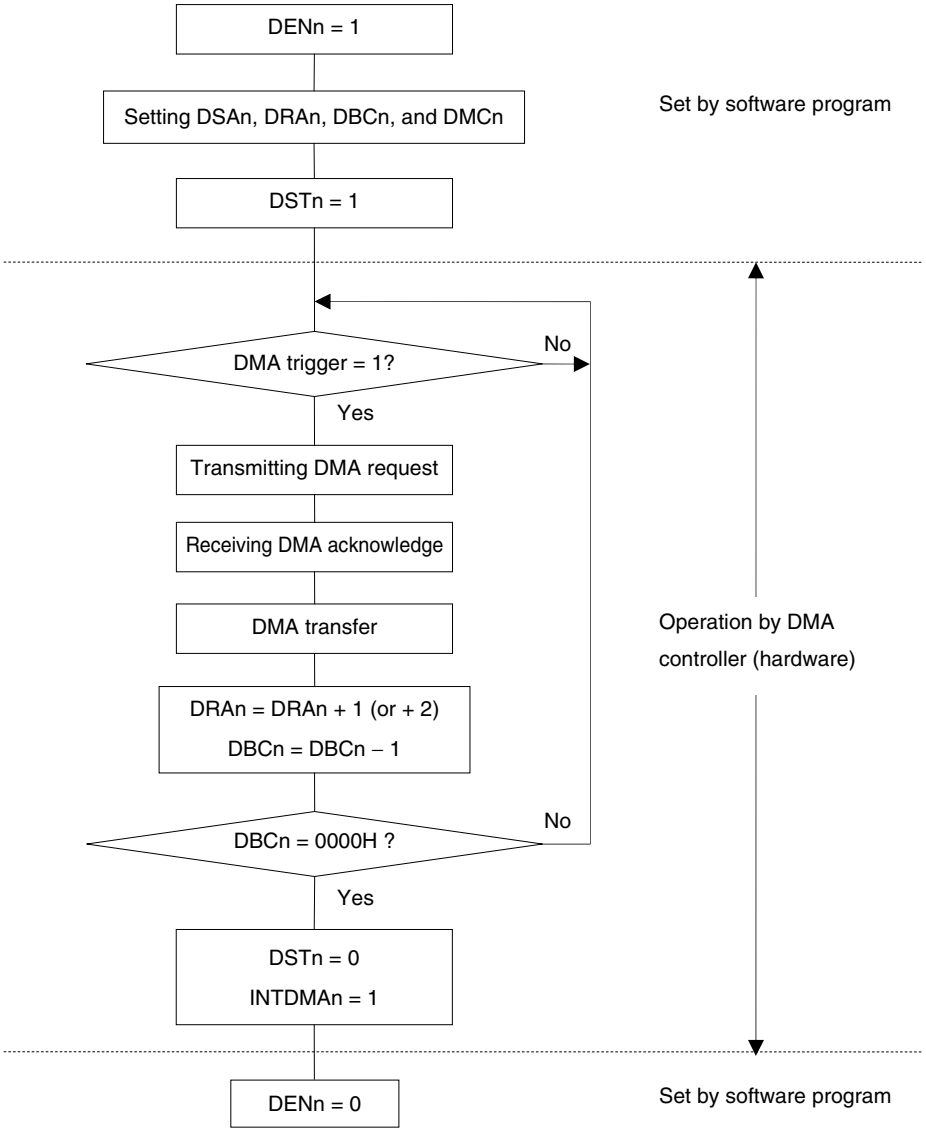
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.

<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 (INTDMAn).

<6> Stop the operation of the DMA controller by clearing DENn 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 DSn) of the DMCn register.

DRSn	DSn	DMA Transfer Mode
0	0	Transfer from SFR of 1-byte data (fixed address) to RAM (address is incremented by +1)
0	1	Transfer from SFR of 2-byte data (fixed address) to RAM (address is incremented by +2)
1	0	Transfer from RAM of 1-byte data (address is incremented by +1) to SFR (fixed address)
1	1	Transfer from RAM of 2-byte data (address is incremented by +2) to SFR (fixed address)

By using these transfer modes, up to 1024 bytes of data can be consecutively transferred by using the serial interface, data resulting from A/D conversion can be consecutively transferred, and port data can be scanned at fixed time intervals by using a timer.

16.4.3 Termination of DMA transfer

When DBCn = 00H and DMA transfer is completed, the DSTn bit is automatically cleared to 0. An interrupt request (INTDMAn) is generated and transfer is terminated.

When the DSTn bit is cleared to 0 to forcibly terminate DMA transfer, the DBCn and DRAn registers hold the value when transfer is terminated.

The interrupt request (INTDMAn) is not generated if transfer is forcibly terminated.

Remark n: DMA channel number (n = 0, 1)

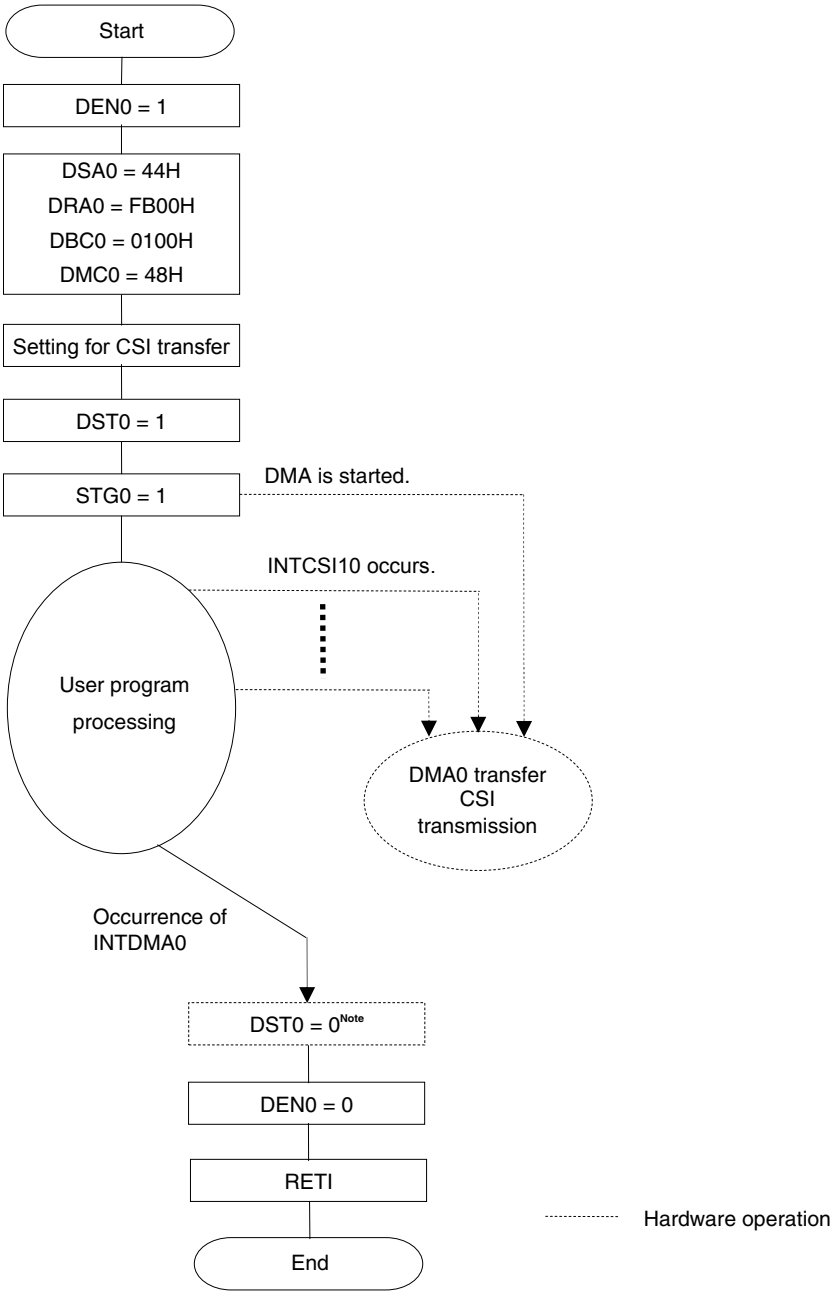
16.5 Example of Setting of DMA Controller

16.5.1 CSI consecutive transmission

A flowchart showing an example of setting for CSI consecutive transmission is shown below.

- Consecutive transmission of CSI10 (256 bytes)
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI10 (software trigger (STG0) only for the first start source)
- Interrupt of CSI10 is specified by IFC03 to IFC00 (bits 3 to 0 of the DMC0 register) = 1000B.
- Transfers FFB00H to FFBFFH (256 bytes) of RAM to FFF44H of the data register (SIO10) of CSI.

Figure 16-7. Example of Setting for 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 **16.5.5 Forced termination by software**).

The fist trigger for consecutive transmission is not started by the interrupt of CSI. In this example, it start by a software trigger.

CSI transmission of the second time and onward is automatically executed.

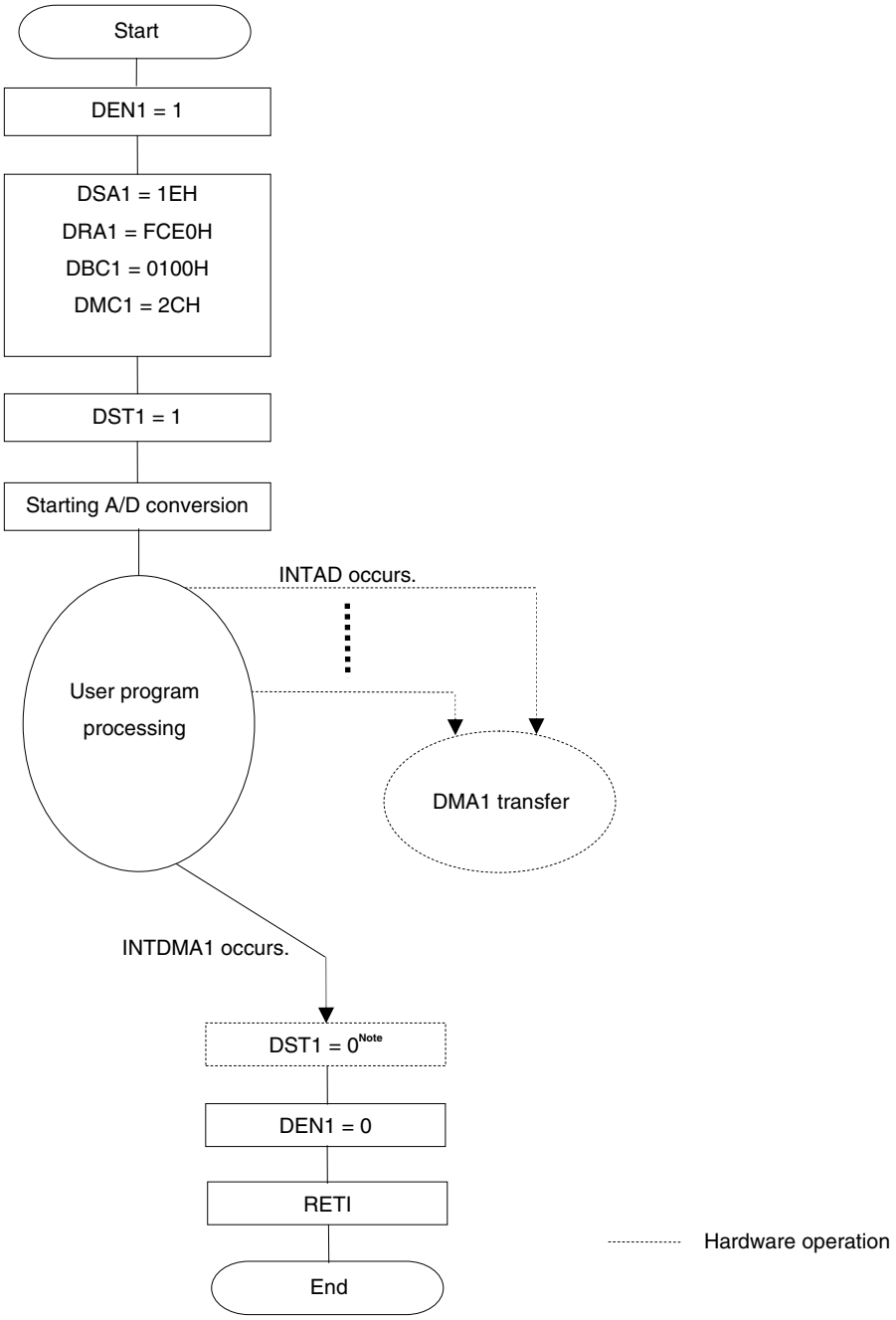
A DMA interrupt (INTDMA0) occurs when the last transmit data has been written to the data register.

16.5.2 Consecutive capturing of A/D conversion results

A flowchart of an example of setting for consecutively capturing A/D conversion results is shown below.

- Consecutive capturing of A/D conversion results.
- DMA channel 1 is used for DMA transfer.
- DMA start source: INTAD
- Interrupt of A/D is specified by IFC13 to IFC10 (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 512 bytes of FFCE0H to FFEDFH of RAM.

Figure 16-8. Example of Setting of Consecutively Capturing A/D Conversion Results



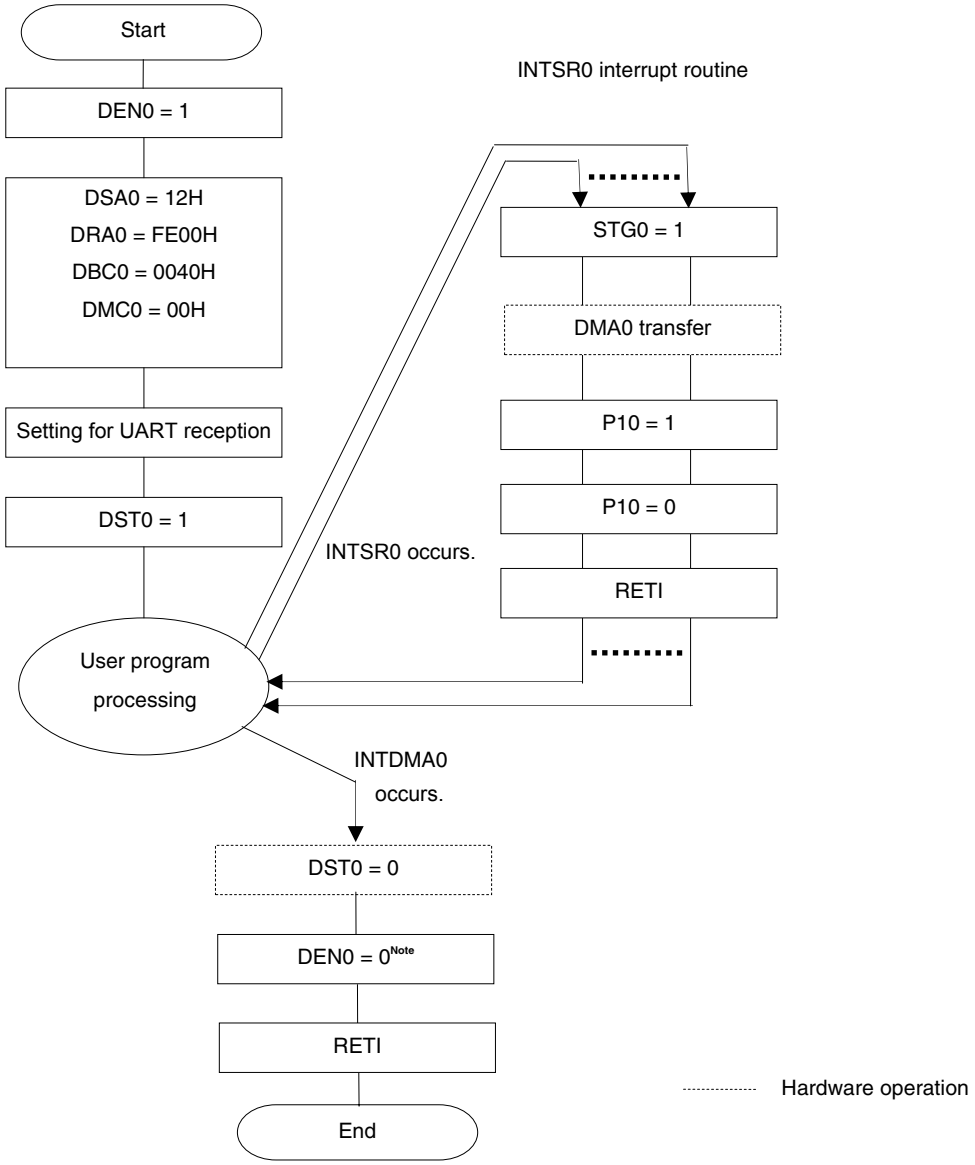
Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set DST1 to 0 and then DEN1 to 0 (for details, refer to **16.5.5 Forced termination by software**).

16.5.3 UART consecutive reception + ACK transmission

A flowchart illustrating an example of setting for UART consecutive reception + ACK transmission is shown below.

- Consecutively receives data from UART0 and outputs ACK to P10 on completion of reception.
- DMA channel 0 is used for DMA transfer.
- DMA start source: Software trigger (DMA transfer on occurrence of an interrupt is disabled.)
- Transfers FFF12H of UART receive data register 0 (RXD0) to 64 bytes of FFE00H to FFE3FH of RAM.

Figure 16-9. Example of Setting for UART Consecutive Reception + ACK Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set DST0 to 0 and then DEN0 to 0 (for details, refer to **16.5.5 Forced termination by software**).

Remark This is an example where a software trigger is used as a DMA start source.
If ACK is not transmitted and if only data is consecutively received from UART, the UART reception end interrupt (INTSR0) can be used to start DMA for data reception.

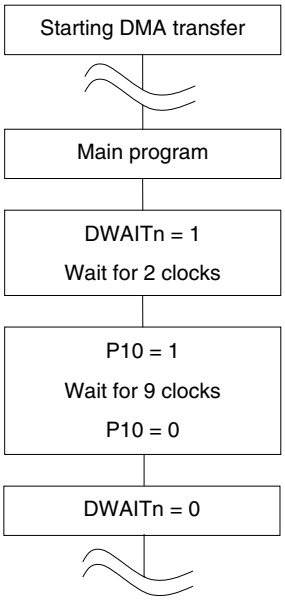
16.5.4 Holding DMA transfer pending by DWAITn

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 system, a DMA transfer can be held pending by setting the DWAITn bit to 1. The DMA transfer for a transfer trigger that occurred while DMA transfer was held pending is executed after the pending status is canceled. However, because only one transfer trigger can be held pending for each channel, even if multiple transfer triggers occur for one channel during the pending status, only one DMA transfer is executed after the pending status is canceled.

To output a pulse with a width of 10 clocks of the operating frequency from the P10 pin, for example, the clock width increases to 12 if a DMA transfer is started midway. In this case, the DMA transfer can be held pending by setting DWAITn to 1.

After setting DWAITn to 1, it takes two clocks until a DMA transfer is held pending.

Figure 16-10. Example of Setting for Holding DMA Transfer Pending by DWAITn



Caution When DMA transfer is held pending while using both DMA channels, be sure to held the DMA transfer pending for both channels (by setting DWAIT0 and DWAIT1 to 1). If the DMA transfer of one channel is executed while that of the other channel is held pending, DMA transfer might not be held pending for the latter channel.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

16.5.5 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>

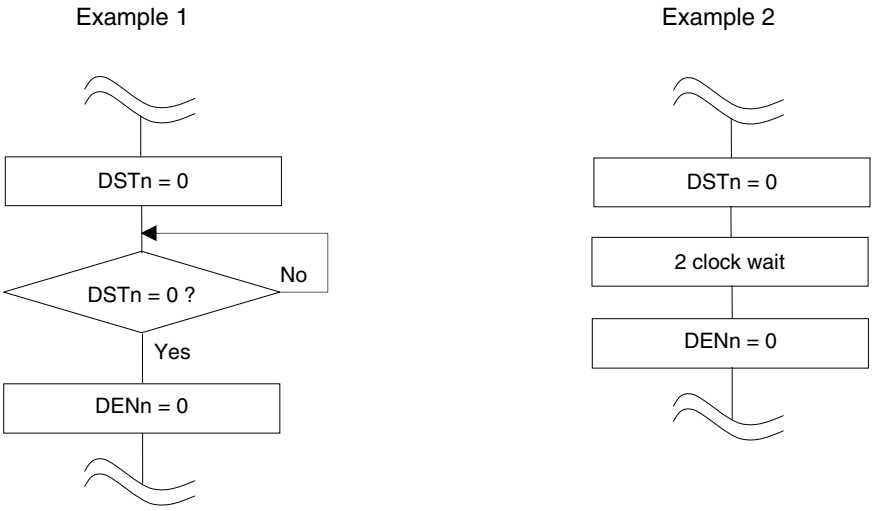
- Set DSTn to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software, confirm by polling that DSTn has actually been cleared to 0, and then set DENn to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction).

- 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.

<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 16-11. Forced Termination of DMA Transfer (1/2)



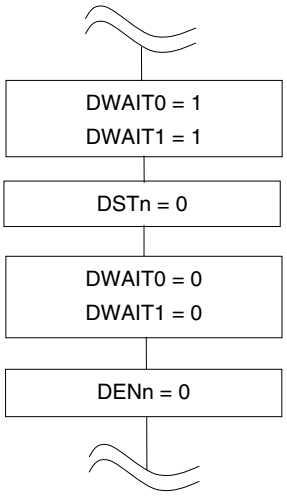
Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

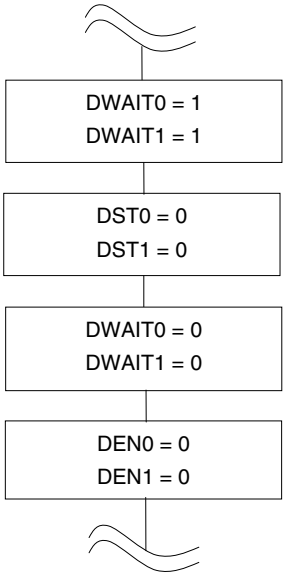
Figure 16-11. Forced Termination of DMA Transfer (2/2)

Example 3

- Procedure for forcibly terminating the DMA transfer for one channel if both channels are used



- Procedure for forcibly terminating the DMA transfer for both channels if both channels are used



Caution In example 3, the system is not required to wait two clock cycles after the DWAITn bit is set to 1. In addition, the system does not have to wait two clock cycles after clearing the DSTn bit to 0, because more than two clock cycles elapse from when the DSTn bit is cleared to 0 to when the DENn bit is cleared to 0.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

16.6 Cautions on Using DMA Controller

(1) Priority of DMA

During DMA transfer, a request from the other DMA channel is held pending even if generated. The pending DMA transfer is started after the ongoing DMA transfer is completed. If two DMA requests are generated at the same time, however, DMA channel 0 takes priority over DMA channel 1.

If a DMA request and an interrupt request are generated at the same time, the DMA transfer takes precedence, and then interrupt servicing is executed.

(2) DMA response time

The response time of DMA transfer is as follows.

Table 16-2. Response Time of DMA Transfer

	Minimum Time	Maximum Time
Response time	3 clocks	10 clocks[Note]

Note The maximum time necessary to execute an instruction from internal RAM is 16 clock cycles.

Cautions 1. The above response time does not include the two clock cycles required for a DMA transfer.

2. When executing a DMA pending instruction (see 16.6 (4)), the response time is extended by the execution time of the instruction to be held pending.

3. Do not specify successive transfer triggers for a channel within a period equal to the maximum response time plus one clock cycle, because they might be ignored.

Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

(3) Operation in standby mode

The DMA controller operates as follows in the standby mode.

Table 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.

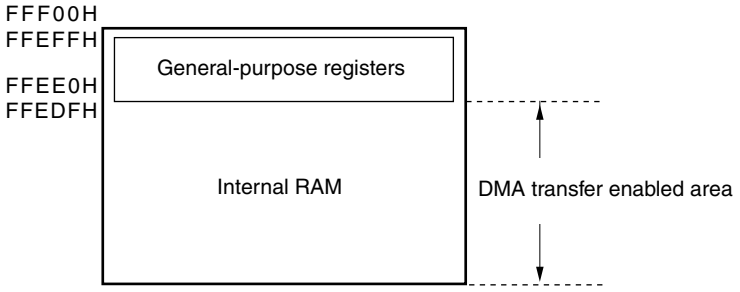
(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified

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: 9, internal: 39

(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/KG3-C has a total of 49 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[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	0	INTWDTI	Watchdog timer interval[Note 3] (75% of overflow time)	Internal	0004H	(A)
	1	INTLVI	Low-voltage detection[Note 4]		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	INTDA	End of CEC 1-byte communication	Internal	0014H	(A)
	9	INTCE	End of CEC communication		0016H	
	10	INTERR	CEC 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	UART1 reception transfer end		0026H	
	18	INTSRE1	UART1 reception communication error occurrence		0028H	
	19	INTIICA	End of IICA 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	
	22	INTTM02	End of timer array unit 0 channel 2 count or capture		0030H	
	23	INTTM03	End of timer array unit 0 channel 3 count or capture		0032H	

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 47 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[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	24	INTAD	End of A/D conversion	Internal	0034H	(A)
	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	003AH	(C)
	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	INTRERR3	Remote control channel 3 error occurrence		003EH	
	30	INTRA3	Remote control channel 3 guide pulse detection, read request for 8-bit shift data, or data reception completion		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	INTSR2	UART2 reception transfer end		004AH	
	36	INTP7	Pin input edge detection	External	004CH	(B)
	37	INTP8			004EH	
	38	INTRERR2	Remote control channel 2 error occurrence	Internal	0050H	(A)
	39	INTRA2	Remote control channel 2 guide pulse detection, read request for 8-bit shift data, or data reception completion		0052H	
	40	INTRERR1	Remote control channel 1 error occurrence		0054H	
	41	INTTM10	End of timer array unit 1 channel 0 count or capture		0056H	
	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	INTSRE2	UART2 reception communication error occurrence		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 47 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[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
Maskable	45	INTRA1	Remote control channel 1 guide pulse detection, read request for 8-bit shift data, or data reception completion	Internal	005EH	(A)
	46	INTRERR0	Remote control channel 0 error occurrence		0060H	
	47	INTRA0	Remote control channel 0 guide pulse detection, read request for 8-bit shift data, or data reception completion		0062H	
Software	–	BRK	Execution of BRK instruction	–	007EH	(D)
Reset	–	RESET	$\overline{\text{RESET}}$ pin input	–	0000H	–
		POC	Power-on-clear			
		LVI	Low-voltage detection[Note 3]			
		WDT	Overflow of watchdog timer			
		TRAP	Execution of illegal instruction[Note 4]			

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 47 indicates the lowest priority.

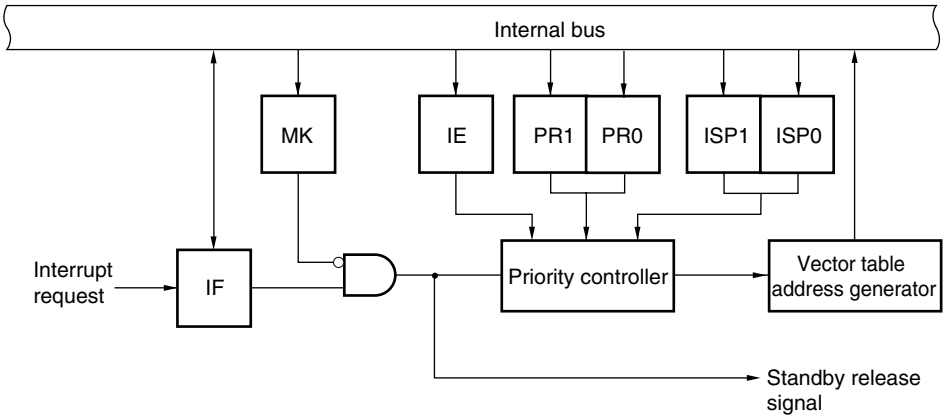
2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 17-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

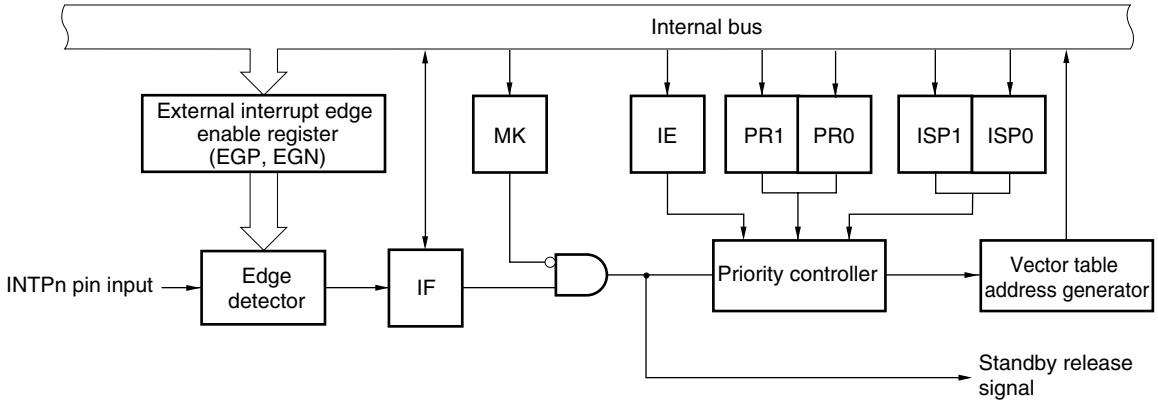
4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 17-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)

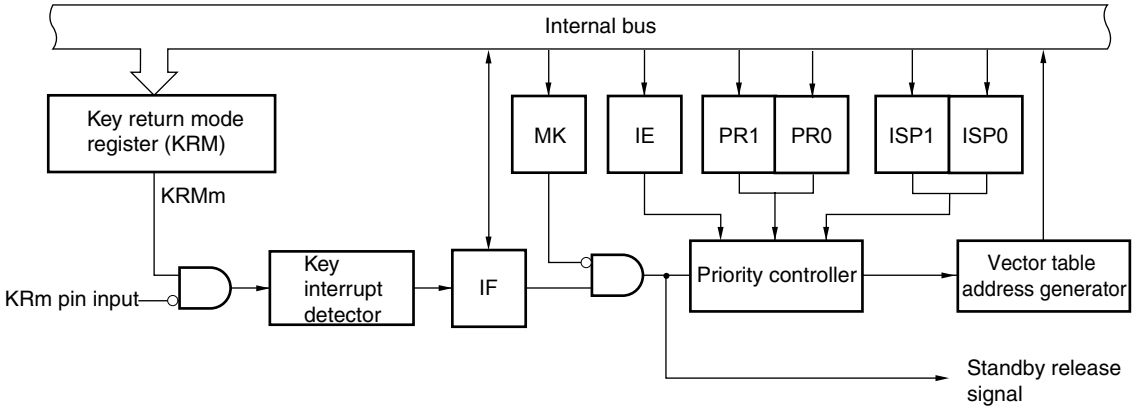


Remarks 1. IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

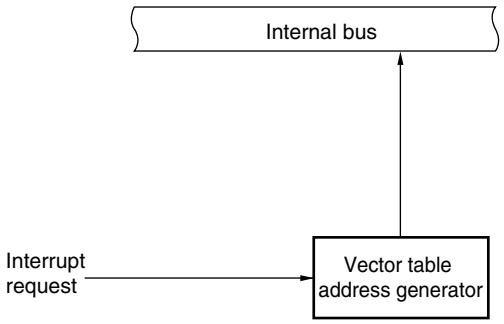
2. n = 0 to 5, 7, 8

Figure 17-1. Basic Configuration of Interrupt Function (2/2)

(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
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1	
INTP0	PIF0		PMK0		PPR00, PPR10	
INTP1	PIF1		PMK1		PPR01, PPR11	
INTP2	PIF2		PMK2		PPR02, PPR12	
INTP3	PIF3		PMK3		PPR03, PPR13	
INTP4	PIF4		PMK4		PPR04, PPR14	
INTP5	PIF5		PMK5		PPR05, PPR15	
INTDA	DAIF	IF0H	DAMK	MK0H	DAPR0, DAPR1	PR00H, PR10H
INTCE	CEIF		CEMK		CEPR0, CEPR1	
INTERR	ERRIF		ERRMK		ERRPR0, ERRPR1	
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10	
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11	
INTST0 Note 1	STIF0 Note 1		STMK0 Note 1		STPR00, STPR10 Note 1	
INTCSI00 Note 1	CSIIF00 Note 1		CSIMK00 Note 1		CSIPR000, CSIPR100 Note 1	
INTSR0 Note 2	SRIF0 Note 2		SRMK0 Note 2		SRPR00, SRPR10 Note 2	
INTCSI01 Note 2	CSIIF01 Note 2		CSIMK01 Note 2		CSIPR001, CSIPR101 Note 2	
INTSRE0	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 three 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 three interrupt sources.

Table 17-2. Flags Corresponding to Interrupt Request Sources (2/2)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register
INTST1[Note 1]	STIF1[Note 1]	IF1L	STMK1[Note 1]	MK1L	STPR01, STPR11[Note 1]	PR01L, PR11L
INTCSI10[Note 1]	CSIIF10[Note 1]		CSIMK10[Note 1]		CSIPR010, CSIPR110[Note 1]	
INTIIC10[Note 1]	IICIF10[Note 1]		IICMK10[Note 1]		IICPR010, IICPR110[Note 1]	
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11	
INTSRE1	SREIF1		SREMK1		SREPR01, SREPR11	
INTIICA	IICAIF		IICAMK		IICAPR0, IICAPR1	
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102	
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103	
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1	
INTRTCI	RTCIIF		RTCIMK		RTCIPR0, RTCIPR1	
INTKR	KRIF		KRMK		KRPR0, KRPR1	
INTST2[Note 2]	STIF2[Note 2]		STMK2[Note 2]		STPR02, STPR12[Note 2]	
INTCSI20[Note 2]	CSIIF20[Note 2]		CSIMK20[Note 2]		CSIPR020, CSIPR120[Note 2]	
INTIIC20[Note 2]	IICIF20[Note 2]		IICMK20[Note 2]		IICPR020, IICPR120[Note 2]	
INTRERR3	ERRIF3		ERRMK3		ERRPR03, ERRPR13	
INTRA3	RAIF3		RAMK3		RAPR03, RAPR13	
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104	
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106	
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107	
INTSR2	SRIF2		SRMK2		SRPR02, SRPR12	
INTP7	PIF7		PMK7		PPR07, PPR17	
INTP8	PIF8		PMK8		PPR08, PPR18	
INTRERR2	ERRIF2		ERRMK2		ERRPR02, ERRPR12	
INTRA2	RAIF2		RAMK2		RAPR02, RAPR12	
INTRERR1	ERRIF1	IF2H	ERRMK1	MK2H	ERRPR01, ERRPR11	PR02H, PR12H
INTTM10	TMIF10		TMMK10		TMPR010, TMPR110	
INTTM11	TMIF11		TMMK11		TMPR011, TMPR111	
INTTM12	TMIF12		TMMK12		TMPR012, TMPR112	
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12	
INTRA1	RAIF1		RAMK1		RAPR01, RAPR11	
INTRERR0	ERRIF0		ERRMK0		ERRPR00, ERRPR10	
INTRA0	RAIF0		RAMK0		RAPR00, RAPR10	

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 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.

(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	LVIIF	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	ERRIF	CEIF	DAIF

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	IICAIF	SREIF1	SRIF1	STIF1 CSIIF10 IICIF10

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	TMIF04	RAIF3	ERRIF3	STIF2 CSIIF20 IICIF20	KRIF	RTCIIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	RAIF12	ERRIF2	PIF8	PIF7	SRIF2	TMIF07	TMIF06	TMIF05

Figure 17-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (2/2)

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2H	RAIF0	ERRIF0	RAIF1	SREIF2	TMIF12	TMIF11	TMIF10	ERRIF1

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. 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.

2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of 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	ERRMK	CEMK	DAMK

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	IICAMK	SREMK1	SRMK1	STMK1 CSIMK10 IICMK10

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	TMMK04	RAMK3	ERRMK3	STMK2 CSIMK20 IICMK20	KRMK	RTCIMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	RAMK2	ERRMK2	PMK8	PMK7	SRMK2	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	RAMK0	ERRMK0	RAMK1	SREMK2	TMMK12	TMMK11	TMMK10	ERRMK1

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

(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 CSIPR001	STPR00 CSIPR000	DMAPR01	DMAPR00	ERRPR0	CEPR0	DAPR0

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10	SRPR10 CSIPR101	STPR10 CSIPR100	DMAPR11	DMAPR10	ERRPR1	CEPR1	DAPR1

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICAPR0	SREPR01	SRPR01	STPR01 CSIPR010 IICPR010

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICAPR1	SREPR11	SRPR11	STPR11 CSIPR110 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	RAPR03	ERRPR03	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	RAPR13	ERRPR13	STPR12 CSIPR120 IICPR120	KRPR1	RTCIPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	RAPR02	ERRPR02	PPR08	PPR07	PPR06	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	RAPR12	ERRPR12	PPR18	PPR17	PPR16	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02H	RAPR00	ERRPR00	RAPR01	SREPR02	TMPR012	TMPR011	TMPR010	ERRPR01

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12H	RAPR10	ERRPR10	RAPR11	SREPR12	TMPR112	TMPR111	TMPR110	ERRPR11

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)

(4) External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)

These registers specify the valid edge for INTP0 to INTP5, INTP7, and INTP8.

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	0	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	0	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	0	0	0	EGP8

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	0	0	0	EGN8

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 5, 7, 8)
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 Bit		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P50 (78K0R/KF3-C) P46 (78K0R/KG3-C)	INTP1
EGP2	EGN2	P51 (78K0R/KF3-C) P47 (78K0R/KG3-C)	INTP2
EGP3	EGN3	P30	INTP3
EGP4	EGN4	P31	INTP4
EGP5	EGN5	P16	INTP5
EGP7	EGN7	P55 (78K0R/KF3-C) P141 (78K0R/KG3-C)	INTP7
EGP8	EGN8	P74	INTP8

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 5, 7, 8

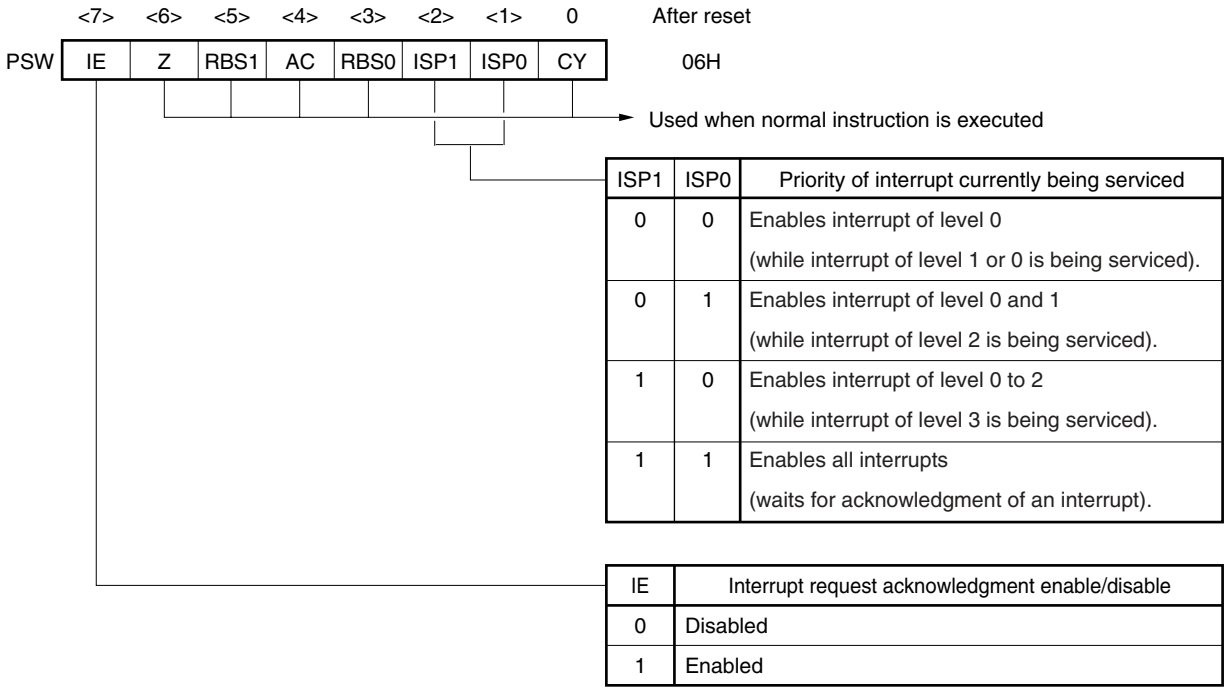
(5) Program status word (PSW)

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



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

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[Note]
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: $1/f_{CLK}$ (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

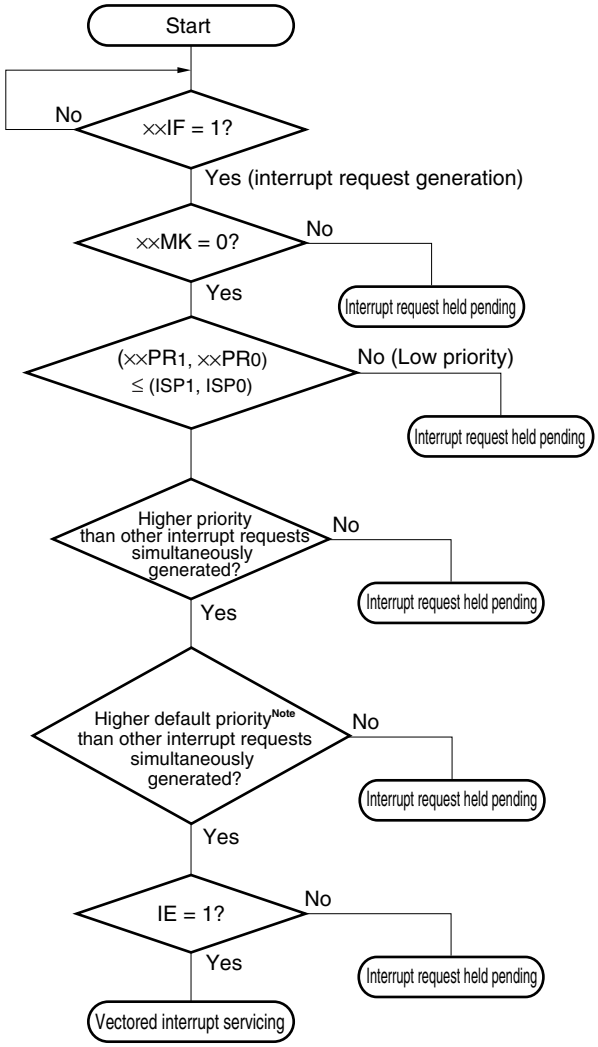
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 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 ISP0 flags. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

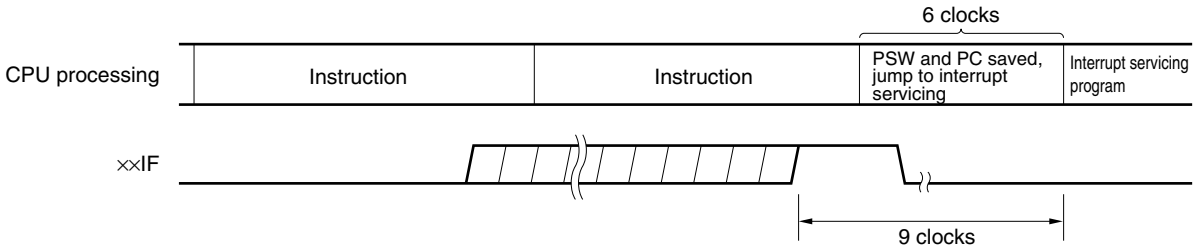
Figure 17-7. Interrupt Request Acknowledgment Processing Algorithm



××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR0: Priority specification flag 0
××PR1: Priority specification flag 1
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1: Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 17-6**)

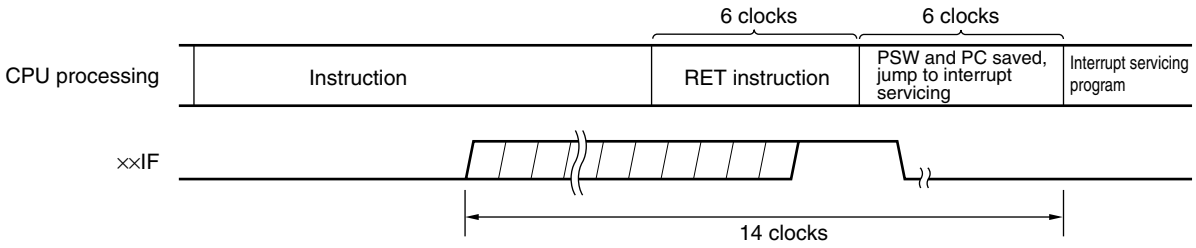
Note For the default priority, refer to **Table 17-1 Interrupt Source List**.

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. If the IE flag is set (1) during level 0 interrupt, other level 0 interrupts are also enabled.

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.

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 1. ○: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

3. ISP0, ISP1, and IE are flags contained in the PSW.

ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.

ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.

ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.

ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment.

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

4. PR is a flag contained in PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)

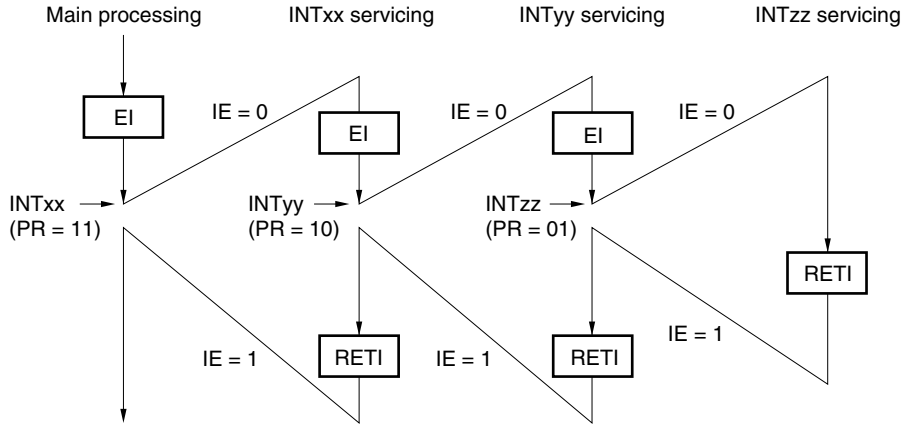
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

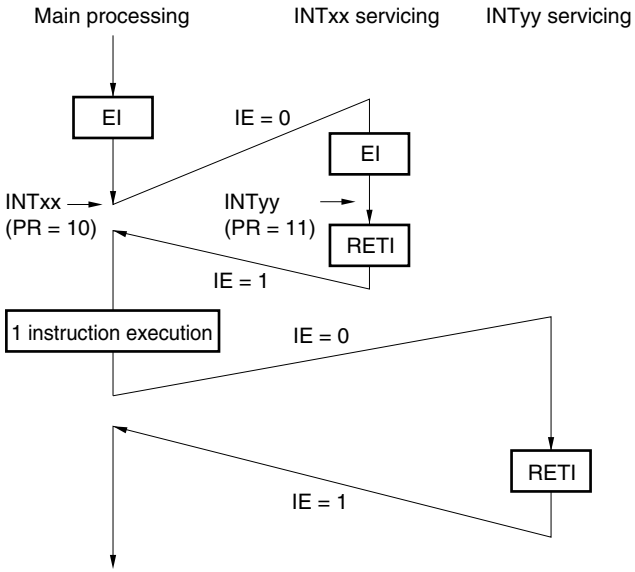
Figure 17-10. Examples of Multiple Interrupt Servicing (1/2)

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

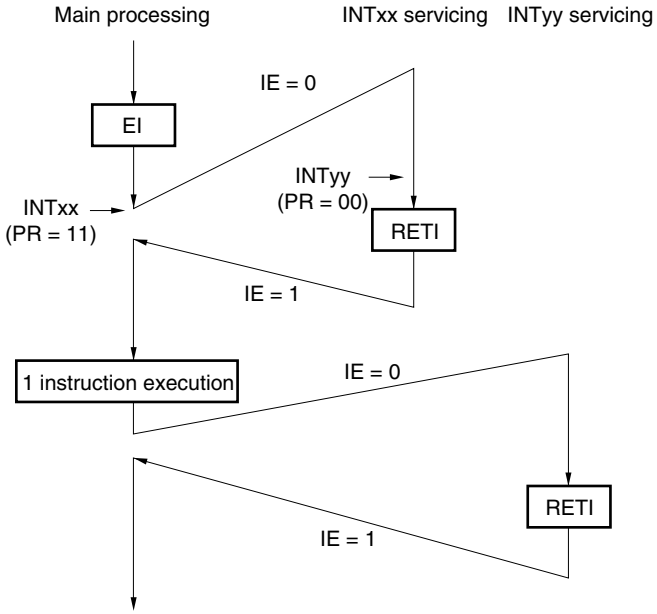


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

Figure 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 ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

17.4.4 Interrupt request hold

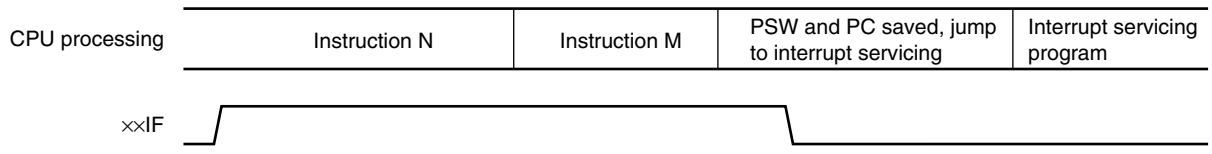
There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- Manipulation instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, 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 ××PR (priority level) values do not affect the operation of ××IF (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.

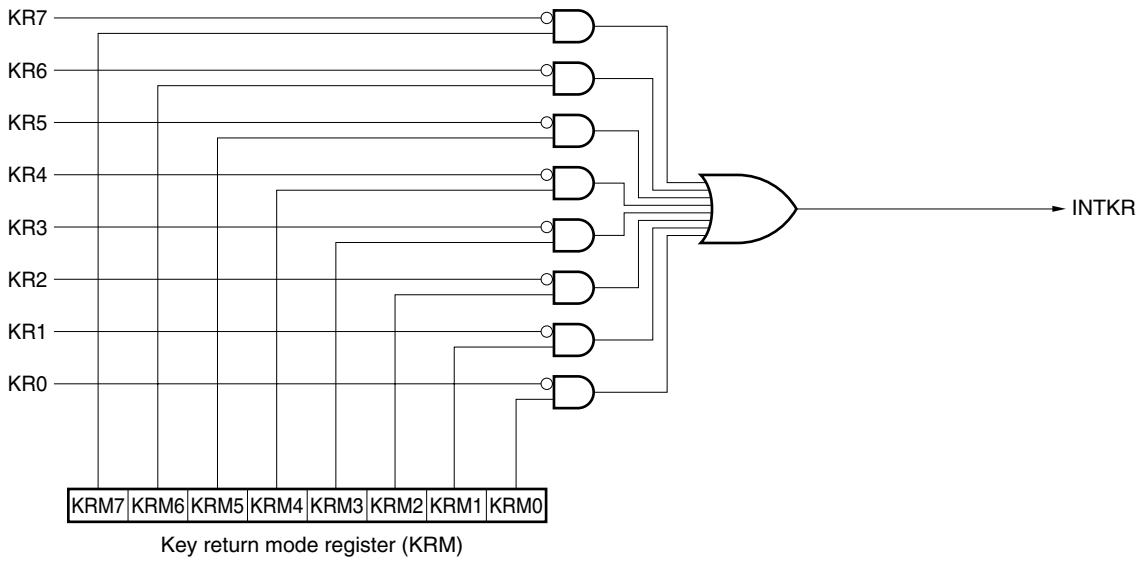
18.2 Configuration of Key Interrupt

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)

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

(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 to P73/KR3, P74/KR4/INTP8, and P75/KR5 to P77/KR7 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, 20 MHz 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.

5. The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.

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 5 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	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	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	102.4 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	204.8 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	819.2 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	26.21 ms min.	13.11 ms min.

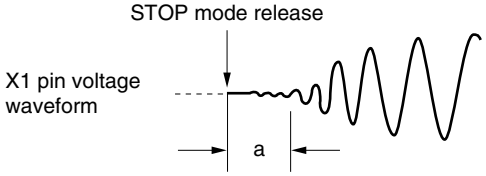
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.

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

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

(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 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	Setting prohibited
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_X$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_X$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.21 ms	13.11 ms

Cautions 1. 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 μs or less is prohibited.

3. Before changing the setting of the OSTS register, confirm that the count operation of the OSTC register is completed.

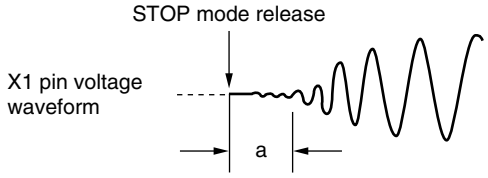
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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 ≤ 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, 20 MHz 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 / Item			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}) or 20 MHz Internal High-Speed Oscillation Clock (f_{IH20})	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}, f_{IH20}	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 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • WDTON = 1 and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Flash memory			Operation stopped		
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		
Timer array unit TAU			Operable		
Real-time counter (RTC)					
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Operates • WDTON = 1 and WDSTBYON = 0: Stops		
Clock output/buzzer output			Operable		
A/D converter					
Serial array unit (SAU)					
Serial interface (IICA)					
CEC transmission/reception circuit					
Remote controller receiver					
Multiplier/divider					
DMA controller					
Power-on-clear function					
Low-voltage detection function					
External interrupt					
Key interrupt function					

Remark f_{IH}: Internal high-speed oscillation clock
f_{IH20}: 20 MHz 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

Table 19-1. Operating Statuses in HALT Mode (2/2)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock
			When CPU Is Operating on XT1 Clock (f_{XT})
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{IH}, f_{IH20}	Status before HALT mode was set is retained
		f_X	
		f_{EX}	
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • WDTON = 1 and WDSTBYON = 0: Stops
CPU			Operation stopped
Flash memory			Operation stopped (wait state in low power 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
Timer array unit TAU			Operable
Real-time counter (RTC)			
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Operates • WDTON = 1 and WDSTBYON = 0: Stops
Clock output/buzzer output			Operable
A/D converter			Cannot operate
Serial array unit (SAU)			Operable
Serial interface (IICA)			Cannot operate
CEC transmission/reception circuit			Operable
Remote controller receiver			
Multiplier/divider			
DMA controller			
Power-on-clear function			
Low-voltage detection function			
External interrupt			
Key interrupt function			

Remark f_{IH}: Internal high-speed oscillation clock
f_{IH20}: 20 MHz 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

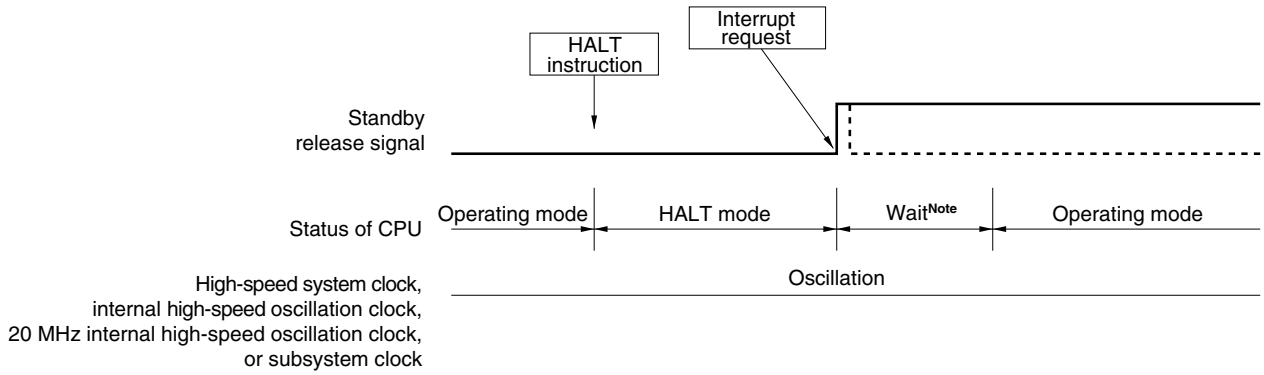
(2) HALT mode release

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

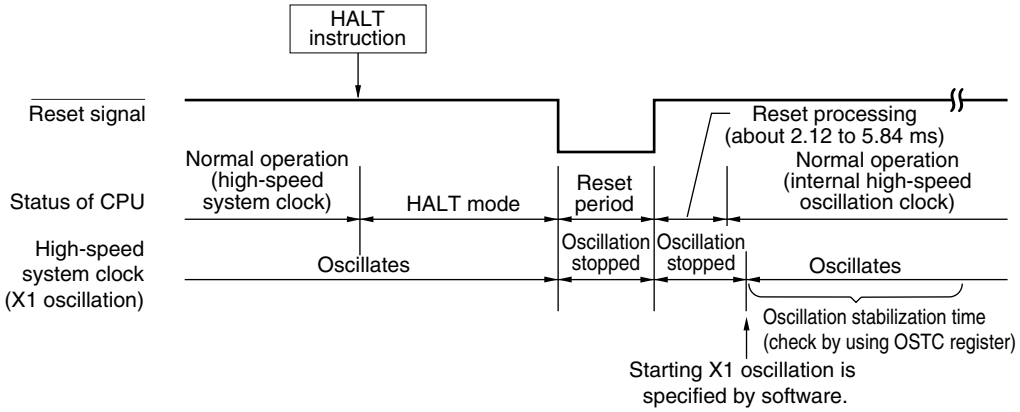
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

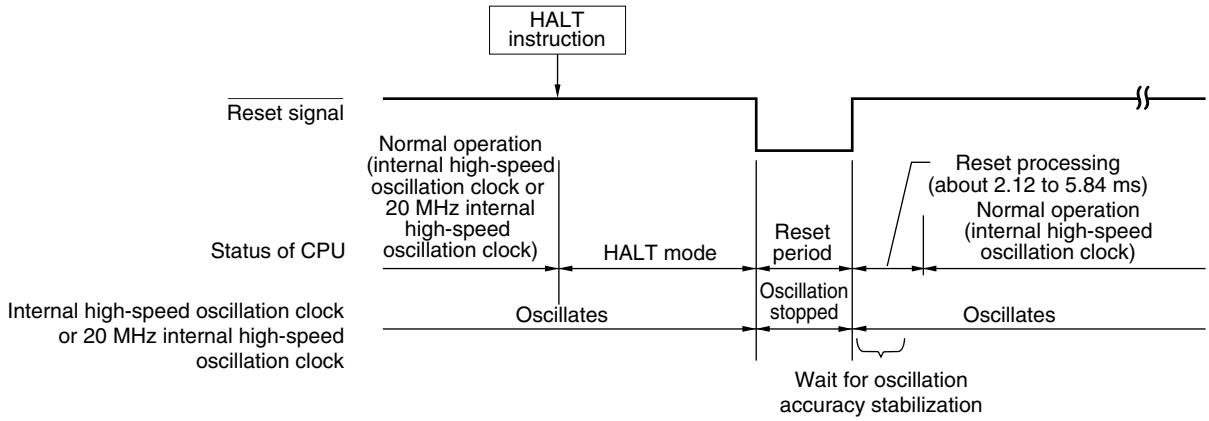
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 19-4. HALT Mode Release by Reset

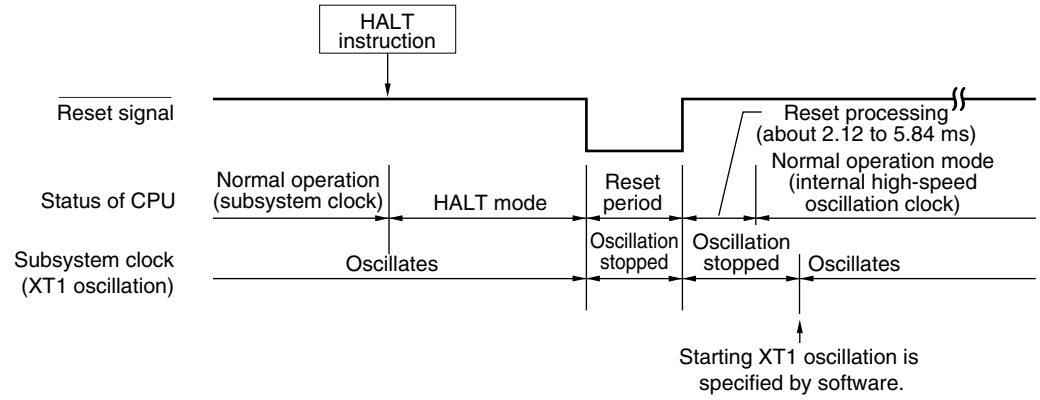
(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock or 20 MHz 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 internal high-speed oscillation clock, X1 clock, or external main system clock.

Cautions 1. Because the interrupt request signal is used to release 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 released 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.

2. The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.

The operating statuses in the STOP mode are shown below.

Table 19-2. Operating Statuses in STOP Mode

STOP Mode Setting / Item			When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on 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 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • WDTON = 1 and WDSTBYON = 0: Stops		
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		
Timer array unit TAU			Operation disabled		
Real-time counter (RTC)			Operable		
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Operates • WDTON = 1 and WDSTBYON = 0: Stops		
Clock output/buzzer output			Operable only when subsystem clock is selected as the count clock		
A/D converter			Operation disabled		
Serial array unit (SAU)					
Serial interface (IICA)			Wakeup by address match operable		
CEC transmission/reception circuit			Operable only when subsystem clock is selected as the operation clock		
Remote controller receiver					
Multiplier/divider			Operation disabled		
DMA controller					
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					
Key interrupt function					

Remark 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

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).

4. The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.

(2) STOP mode release

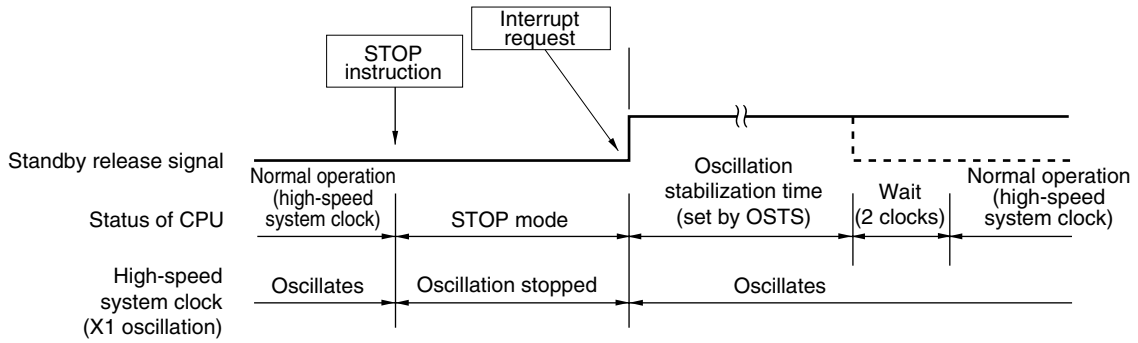
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 19-5. STOP Mode Release by Interrupt Request Generation (1/2)

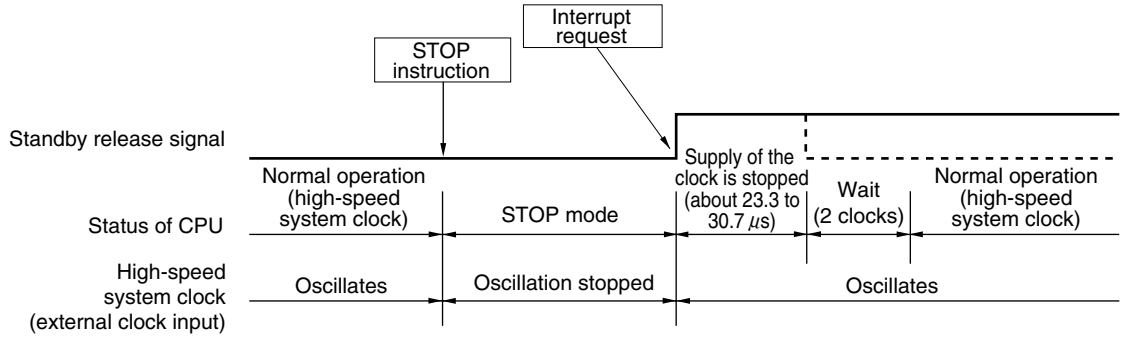
(1) When high-speed system clock (X1 oscillation) is used as CPU clock



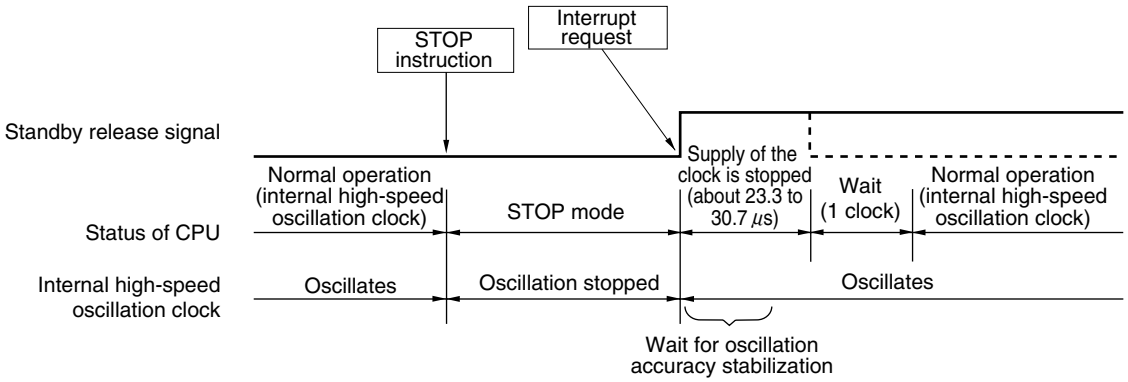
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 19-5. STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (external clock input) is used as CPU clock



(3) When internal high-speed oscillation clock is used as CPU clock



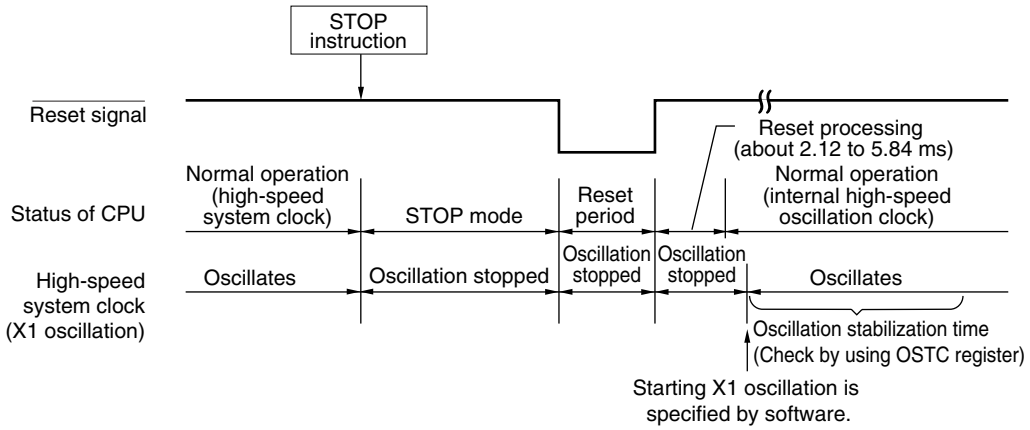
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

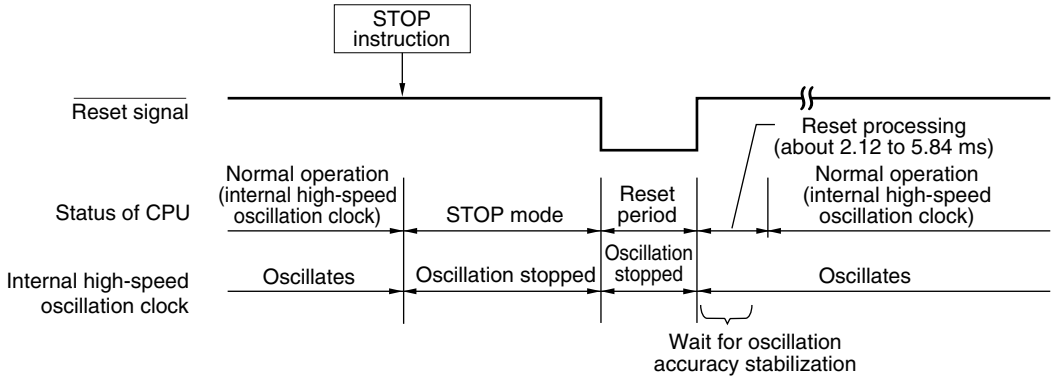
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 19-6. 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 fx: 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 or input voltage (EXLVI) from external input pin, and detection voltage of the low-voltage detector (LVI)
(5) Internal reset by execution of illegal instruction[Note]
(6) Internal reset by a reset processing check error

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[Note], 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 P140, which is low-level output.

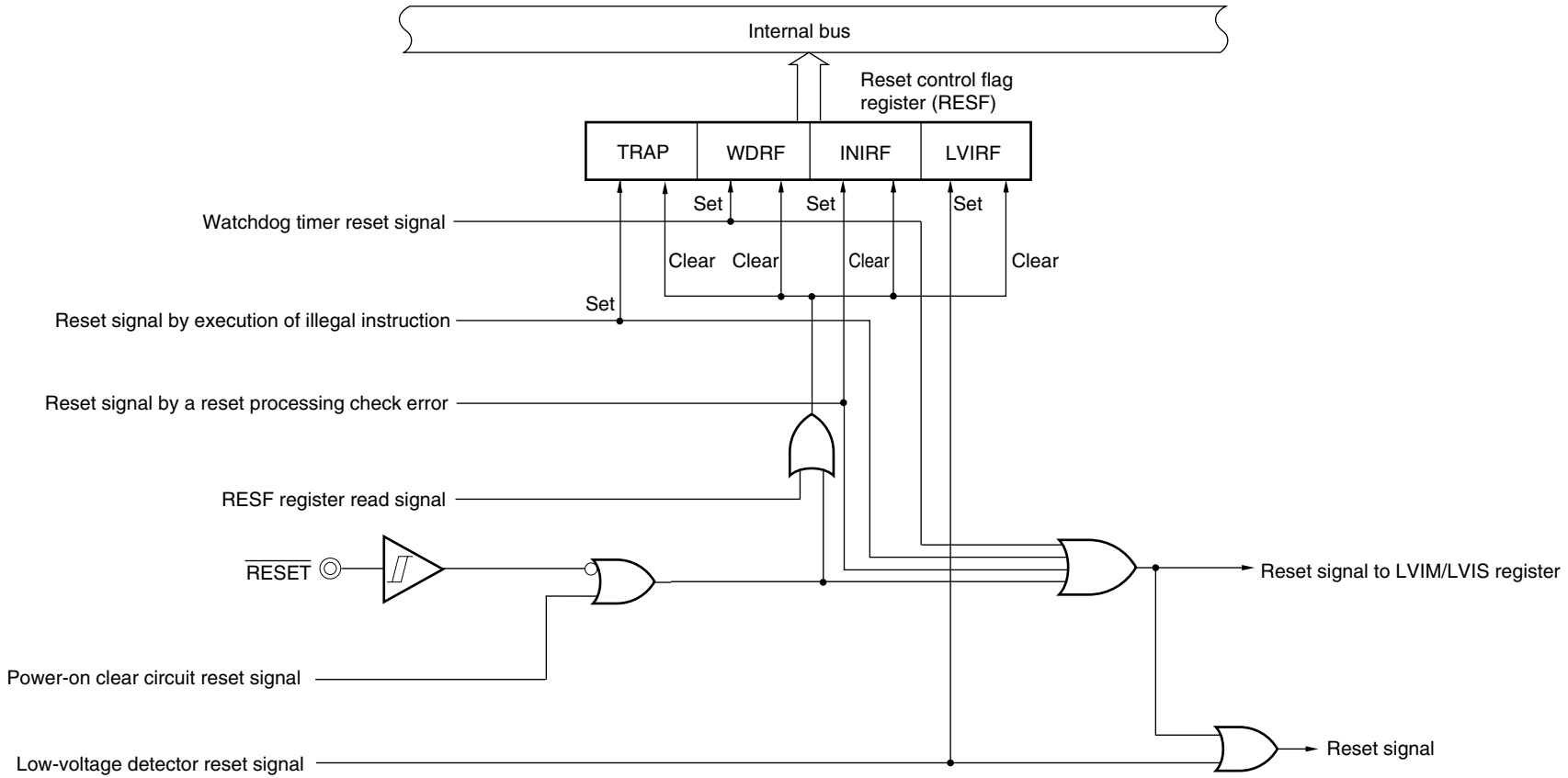
When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the internal high-speed oscillation clock after reset processing. A reset by 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 voltage detection is automatically released when $V_{DD} \geq V_{POR}$ 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 μs or more to the $\overline{\text{RESET}}$ pin.
(To perform an external reset upon power application, a low level of at least 10 μs must be continued during the period in which the supply voltage is within the operating range ($V_{DD} \geq 2.7$ V).)
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.
4. When reset is effected, port pin P130 is set to low-level output and other port pins become high-impedance, because each SFR and 2nd SFR are initialized.

Remark V_{POR}: POC power supply rise detection voltage

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 $\overline{\text{RESET}}$ Input

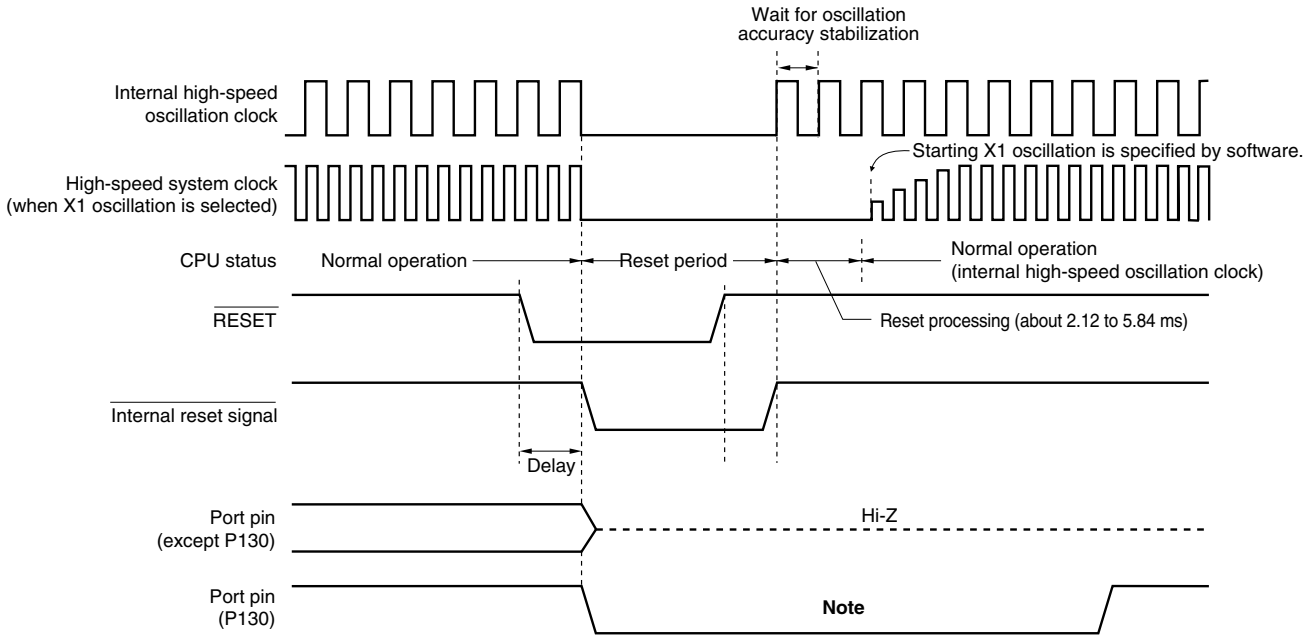
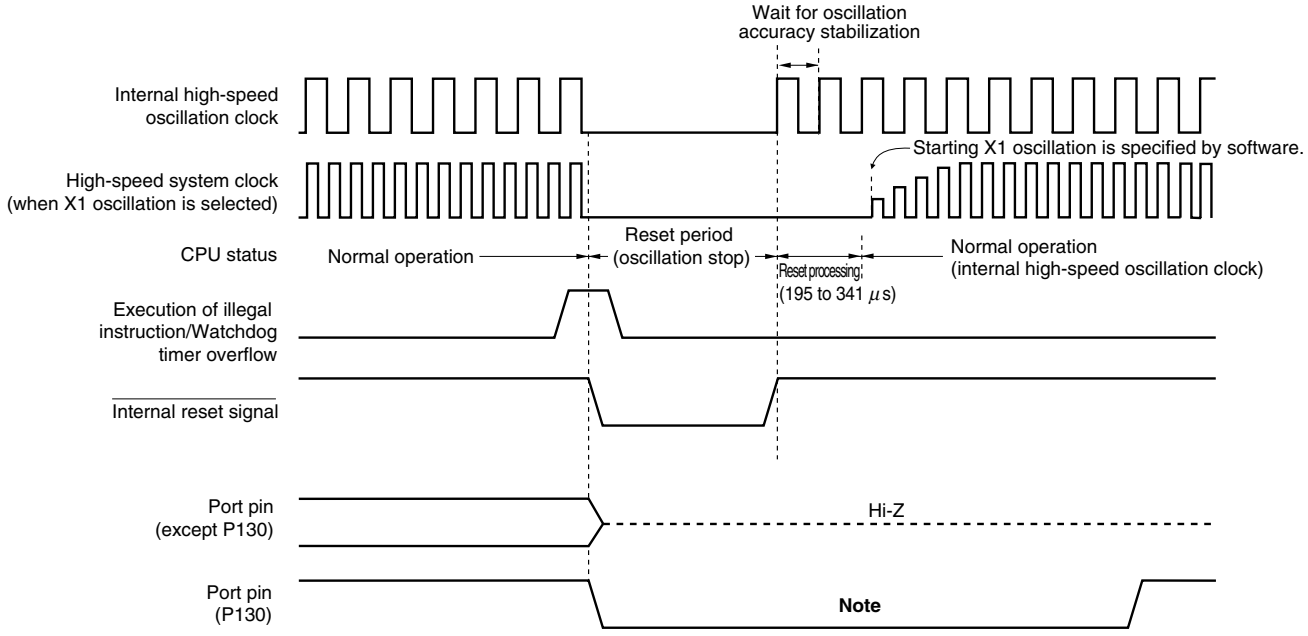


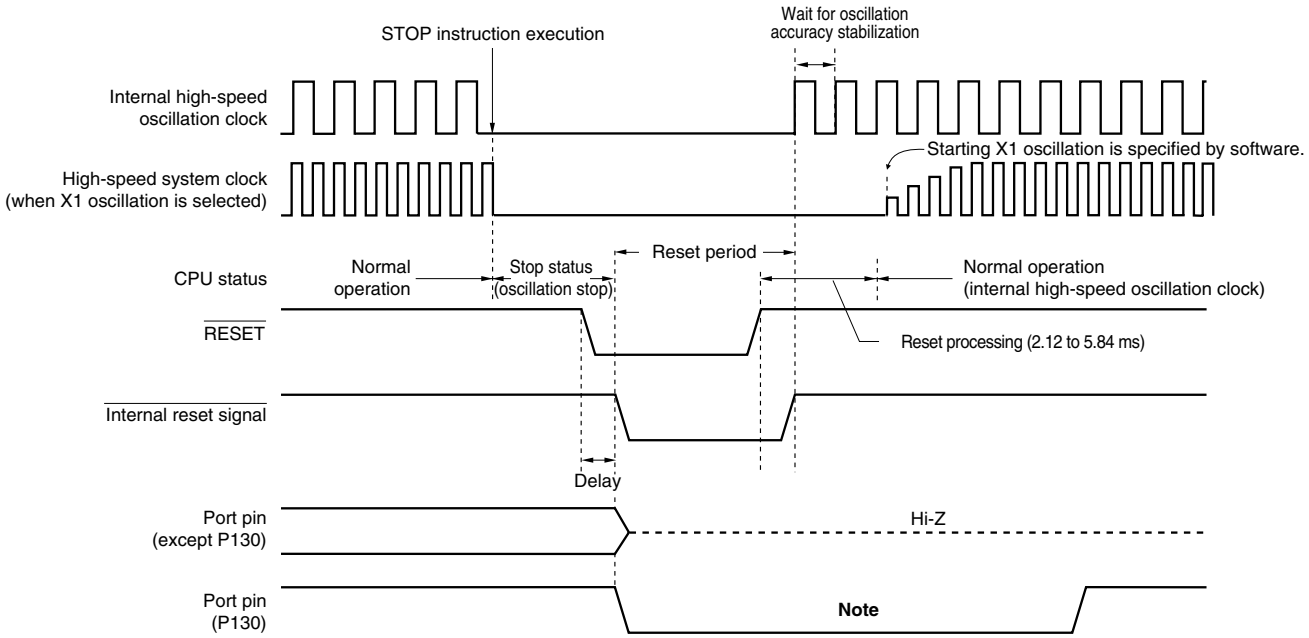
Figure 20-3. Timing of Reset Due to Execution of Illegal Instruction or Watchdog Timer Overflow



Note When P130 is set to high-level output before reset is effected, the output signal of P130 can be dummy-output as a reset signal to an external device, because P130 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 to high-level output by software.

Caution A watchdog timer internal reset also resets the watchdog timer.

Figure 20-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Note When P130 is set to high-level output before reset is effected, the output signal of P130 can be dummy-output as a reset signal to an external device, because P130 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 to high-level output by software.

Remark 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_{IH}, f_{IH20}	Operation stopped
		f_X	Operation stopped (X1 and X2 pins are input port mode)
		f_{EX}	Clock input invalid (pin is input port mode)
	Subsystem clock	f_{XT}	Operation stopped (XT1 and XT2 pins are input port mode)
	f_{IL}		Operation stopped
CPU			
Flash memory			
RAM			Operation stopped (The value, however, is retained when the voltage is at least the power-on-clear detection voltage.)
Port (latch)			Set P130 to low-level output. The port pins except for P130 become high impedance.
Timer array unit TAU			Operation stopped
Real-time counter (RTC)			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Serial array unit (SAU)			
Serial interface (IICA)			
CEC transmission/reception circuit			
Remote controller receiver			
Multiplier/divider			
DMA controller			
Power-on-clear function			Detection operation possible
Low-voltage detection function			Operation stopped (however, operation continues at LVI reset)
External interrupt			Operation stopped
Key interrupt function			

Remark f_{IH}: Internal high-speed oscillation clock
f_{IH20}: 20 MHz internal high-speed oscillation clock
f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{XT}: XT1 oscillation clock
f_{IL}: Internal low-speed oscillation clock

Table 20-2. Hardware Statuses After Reset Acknowledgment (1/6)

Hardware		After Reset Acknowledgment[Note 1]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Processor mode control register (PMC)		00H
Port registers (P0 to P7, P8[Note 3], P9, P11 to P15) (output latches)		00H
Port mode registers	PM0 to PM7, PM8[Note 3], PM9, PM11, PM12, PM14, PM15	FFH
	PM13	FEH
Port input mode registers 0, 1, 6, 14 (PIM0, PIM1, PIM6, PIM14)		00H
Port output mode registers 0, 1, 14 (POM0, POM1, POM14)		00H
Pull-up resistor option registers (PU0, PU1, PU3 to PU7, PU8[Note 3], PU12 to PU14)		00H
Port function registers 6, 11 (PF6, PF11)		00H
Clock operation mode control register (CMC)		00H
Clock operation status control register (CSC)		C0H
System clock control register (CKC)		09H
20 MHz internal high-speed oscillation control register (DSCCTL)		00H
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
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 (TDR00, TDR01, TDR02, TDR03, TDR04, TDR05, TDR06, TDR07, TDR10, TDR11, TDR12)	0000H
	Timer mode registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12 (TMR00, TMR01, TMR02, TMR03, TMR04, TMR05, TMR06, TMR07, TMR10, TMR11, TMR12)	0000H
	Timer status registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12 (TSR00, TSR01, TSR02, TSR03, TSR04, TSR05, TSR06, TSR07, TSR10, TSR11, TSR12)	0000H
	Timer input select registers 0, 1 (TIS0, TIS1)	00H
	Timer counter registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12 (TCR00, TCR01, TCR02, TCR03, TCR04, TCR05, TCR06, TCR07, TCR10, TCR11, TCR12)	FFFFH
	Timer channel enable status register 0 (TE0)	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 channel output registers 0, 1 (TO0, TO1)	0000H
	Timer channel output enable registers 0 , 1(TOE0, TOE1)	0000H
	Timer channel output level registers 0, 1 (TOL0, TOL1)	0000H
	Timer channel 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.

3. 78K0R/KG3-C only

Table 20-2. Hardware Statuses After Reset Acknowledgment (2/6)

Hardware		Status After Reset Acknowledgment[Note 1]
Real-time counter	Sub-count register (RSUBC)	0000H
	Second count register (SEC)	00H
	Minute count register (MIN)	00H
	Hour count register (HOUR)	12H
	Week count register (WEEK)	00H
	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 (ALARMWM)	00H
	Alarm hour register (ALARMWH)	12H
	Alarm week register ALARMWW)	00H
	Control register 0 (RTCC0)	00H
	Control register 1 (RTCC1)	00H
	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[Note 2]
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
Serial array unit (SAU)	Serial data registers 00, 01, 02, 03, 10, 11 (SDR00, SDR01, SDR02, SDR03, SDR10, SDR11)	0000H
	Serial status registers 00, 01, 02, 03, 10, 11 (SSR00, SSR01, SSR02, SSR03, SSR10, SSR11)	0000H
	Serial flag clear trigger registers 00, 01, 02, 03, 10, 11 (SIR00, SIR01, SIR02, SIR03, SIR10, SIR11)	0000H
	Serial mode registers 00, 01, 02, 03, 10, 11 (SMR00, SMR01, SMR02, SMR03, SMR10, SMR11)	0020H
	Serial communication operation setting registers 00, 01, 02, 03, 10, 11 (SCR00, SCR01, SCR02, SCR03, SCR10, SCR11)	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

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 (3/6)

Hardware		Status After Reset Acknowledgment[Note]
Serial interface IICA	IICA shift register (IICA)	00H
	IICA status register (IICS)	00H
	IICA flag register (IICF)	00H
	IICA control register 0 (IICCTL0)	00H
	IICA control register 1 (IICCTL1)	00H
	IICA low-level width setting register (IICWL)	FFH
	IICA high-level width setting register (IICWH)	FFH
	Slave address register (SVA)	00H
CEC transmission/reception circuit	CEC transmission buffer register (CTXD)	00H
	CEC reception buffer register (CRXD)	00H
	CEC communication error status register (CECES)	00H
	CEC communication status register (CECS)	00H
	CEC communication error flag clear trigger register (CECFC)	00H
	CEC control register 0 (CECCTL0)	00H
	CEC local address setting register (CADR)	0000H
	CEC control register 1 (CECCTL1)	00H
	CEC transmission start bit width setting register (STATB)	0000H
	CEC transmission start bit low width setting register (STATL)	0000H
	CEC transmission logical 0 low width setting register (LGC0L)	0000H
	CEC transmission logical 1 low width setting register (LGC1L)	0000H
	CEC transmission data bit width setting register (DATB)	0000H
	CEC reception data sampling time setting register (NOMT)	0000H
	CEC reception start bit minimum low width setting register (STATLL)	0000H
	CEC reception start bit maximum low width setting register (STATLH)	0000H
	CEC reception start bit minimum bit width setting register (STATBL)	0000H
	CEC reception start bit maximum bit width setting register (STATBH)	0000H
	CEC reception logical 0 minimum low width setting register (LGC0LL)	0000H
	CEC reception logical 0 maximum low width setting register (LGC0LH)	0000H
	CEC reception logical 1 minimum low width setting register (LGC1LL)	0000H
	CEC reception logical 1 maximum low width setting register (LGC1LH)	0000H
	CEC reception data bit minimum bit width setting register (DATBL)	0000H
	CEC reception data bit maximum bit width setting register (DATBH)	0000H
Remote controller receiver	Remote controller receive data register 0 (RMDR0)	00H
	Remote controller receive counter register 0 (RMSCR0)	00H
	Remote controller receive shift register 0 (RMSR0)	00H
	Remote control reception error bit detection register 0 (RMERBD0)	00H
	Remote control reception error bit detection shift register 0 (RMERBSR0)	00H
	Remote controller receive interrupt status register 0 (RMINTS0)	00H
	Remote controller receive data register 1 (RMDR1)	00H
	Remote controller receive counter register 1 (RMSCR1)	00H
	Remote controller receive shift register 1 (RMSR1)	00H
	Remote control reception error bit detection register 1 (RMERBD1)	00H
	Remote control reception error bit detection shift register 1 (RMERBSR1)	00H
	Remote controller receive interrupt status register 1 (RMINTS1)	00H

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Table 20-2. Hardware Statuses After Reset Acknowledgment (4/6)

Hardware		Status After Reset Acknowledgment[Note]
Remote controller receiver	Remote controller receive data register 2 (RMDR2)	00H
	Remote controller receive counter register 2 (RMSCR2)	00H
	Remote controller receive shift register 2 (RMSR2)	00H
	Remote control reception error bit detection register 2 (RMERBD2)	00H
	Remote control reception error bit detection shift register 2 (RMERBSR2)	00H
	Remote controller receive interrupt status register 2 (RMINTS2)	00H
	Remote controller receive control 1 register 02 (RMCN102)	00H
	Remote controller receive control 2 register 02 (RMCN202)	00H
	Remote controller receive GPLS compare register 02 (RMGPLS02)	00H
	Remote controller receive GPLL compare register 02 (RMGPLL02)	00H
	Remote controller receive GPHS compare register 02 (RMGPHS02)	00H
	Remote controller receive GPHL compare register 02 (RMGPHL02)	00H
	Remote controller receive DLS compare register 02 (RMDLS02)	00H
	Remote controller receive DLL compare register 02 (RMDLL02)	00H
	Remote controller receive DH0S compare register 02 (RMDH0S02)	00H
	Remote controller receive DH0L compare register 02 (RMDH0L02)	00H
	Remote controller receive DH1S compare register 02 (RMDH1S02)	00H
	Remote controller receive DH1L compare register 02 (RMDH1L02)	00H
	Remote controller receive GPBS compare register 02 (RMGPBS02)	0000H
	Remote controller receive GPBL compare register 02 (RMGPBL02)	0000H
	Remote controller receive DB0S compare register 02 (RMDB0S02)	0000H
	Remote controller receive DB0L compare register 02 (RMDB0L02)	0000H
	Remote controller receive DB1S compare register 02 (RMDB1S02)	0000H
	Remote controller receive DB1L compare register 02 (RMDB1L02)	0000H
	Remote controller receive end width select register L02 (RMERL02)	00H
	Remote controller receive end width select register H02 (RMERH02)	00H
	Remote controller receive noise elimination period setting register 02 (RMNCP102)	00H
	Remote controller receive data register 3 (RMDR3)	00H
	Remote controller receive counter register 3 (RMSCR3)	00H
	Remote controller receive shift register 3 (RMSR3)	00H
	Remote control reception error bit detection register 3 (RMERBD3)	00H
	Remote control reception error bit detection shift register 3 (RMERBSR3)	00H
	Remote controller receive interrupt status register 3 (RMINTS3)	00H
	Remote controller receive control 1 register 13 (RMCN113)	00H
	Remote controller receive control 2 register 13 (RMCN213)	00H

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Table 20-2. Hardware Statuses After Reset Acknowledgment (5/6)

Hardware		Status After Reset Acknowledgment[Note]
Remote controller receiver	Remote controller receive GPLS compare register 03 (RMGPLS03)	00H
	Remote controller receive GPLL compare register 03 (RMGPLL03)	00H
	Remote controller receive GPHS compare register 03 (RMGPHS03)	00H
	Remote controller receive GPHL compare register 03 (RMGPHL03)	00H
	Remote controller receive DLS compare register 03 (RMDLS03)	00H
	Remote controller receive DLL compare register 03 (RMDLL03)	00H
	Remote controller receive DH0S compare register 03 (RMDH0S03)	00H
	Remote controller receive DH0L compare register 03 (RMDH0L03)	00H
	Remote controller receive DH1S compare register 03 (RMDH1S03)	00H
	Remote controller receive DH1L compare register 03 (RMDH1L03)	00H
	Remote controller receive GPBS compare register 13 (RMGPBS13)	0000H
	Remote controller receive GPBL compare register 13 (RMGPBL13)	0000H
	Remote controller receive DB0S compare register 13 (RMDB0S13)	0000H
	Remote controller receive DB0L compare register 13 (RMDB0L13)	0000H
	Remote controller receive DB1S compare register 13 (RMDB1S13)	0000H
	Remote controller receive DB1L compare register 13 (RMDB1L13)	0000H
	Remote controller receive end width select register L13 (RMERL13)	00H
	Remote controller receive end width select register H13 (RMERH13)	00H
	Remote controller receive noise elimination period setting register 13 (RMNCP113)	00H
	Remote controller receive data slew control register (RMSW)	00H
Multiplier/divider	Multiplication/division data register A (L) (MDAL)	0000H
	Multiplication/division data register A (H) (MDAH)	0000H
	Multiplication/division data register B (L) (MDBL)	0000H
	Multiplication/division data register B (H) (MDBH)	0000H
	Multiplication/division data register C (L) (MDCL)	0000H
	Multiplication/division data register C (H) (MDCH)	0000H
	Multiplication/division control register (MDUC)	00H
Key interrupt	Key return mode register (KRM)	00H

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Table 20-2. Hardware Statuses After Reset Acknowledgment (6/6)

Hardware		Status After Reset Acknowledgment[Note 1]
Reset function	Reset control flag register (RESF)	Undefined[Note 2]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note 3]
	Low-voltage detection level select register (LVIS)	0EH[Note 2]
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.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by INIRF	Reset by LVI
RESF	TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held
	WDRF bit			Held	Set (1)	Held	Held
	INIRF bit			Held	Held	Set (1)	Held
	LVIRF bit			Held	Held	Held	Set (1)
LVIS		Cleared (0EH)	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/Kx3-C. 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: 00[Note 1] 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[Note 2]
0	Internal reset request is not generated, or RESF register 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 register is cleared.
1	Internal reset request is generated.

INIRF	Internal reset request by a reset processing check error
0	Internal reset request is not generated, or RESF register 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 register is cleared.
1	Internal reset request is generated.

Notes 1. The value after reset varies depending on the reset source.

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

Cautions 1. Do not read data by a 1-bit memory manipulation instruction.

2. When 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

Flag \ Reset Source	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by INIRF	Reset by LVI
TRAP	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held
WDRF			Held	Set (1)	Held	Held
INIRF			Held	Held	Set (1)	Held
LVIRF			Held	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 1.61 V ±0.09 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 V ±0.2 V.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 V), generates internal reset signal when $V_{DD} < V_{PDR}$.

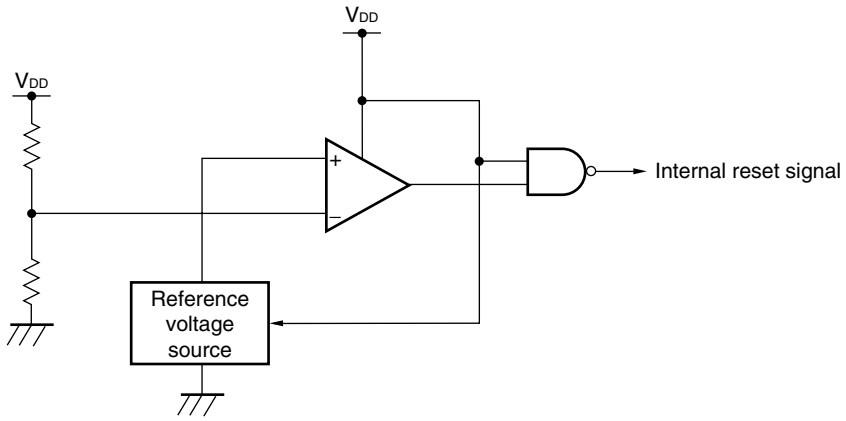
  Caution If an internal reset signal is generated in the POC circuit, TRAP, WDRF, and LVIRF of the reset control flag register (RESF) is cleared.

  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_{PDR} = 1.61 V ±0.09 V), the reset status is released.

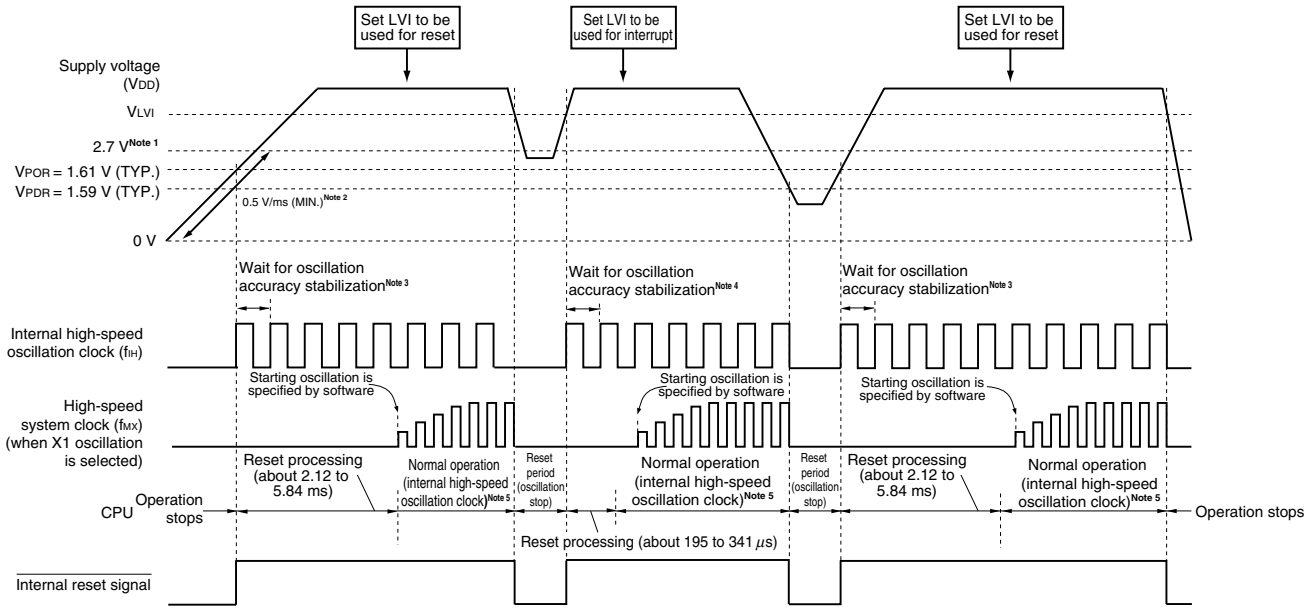
  Caution If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage (V_{DD}) exceeds 2.07 V ±0.2 V.

- The supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 V) are compared. When $V_{DD} < V_{PDR}$, 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 2.7 V ≤ V_{DD} ≤ 5.5 V. To make the state at lower than 2.7 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 2.7 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 2.7 V, or set LVI to ON by default by using an option byte (option byte: LVIOFF = 0).

3. The reset processing time, such as when waiting for internal voltage stabilization, 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.

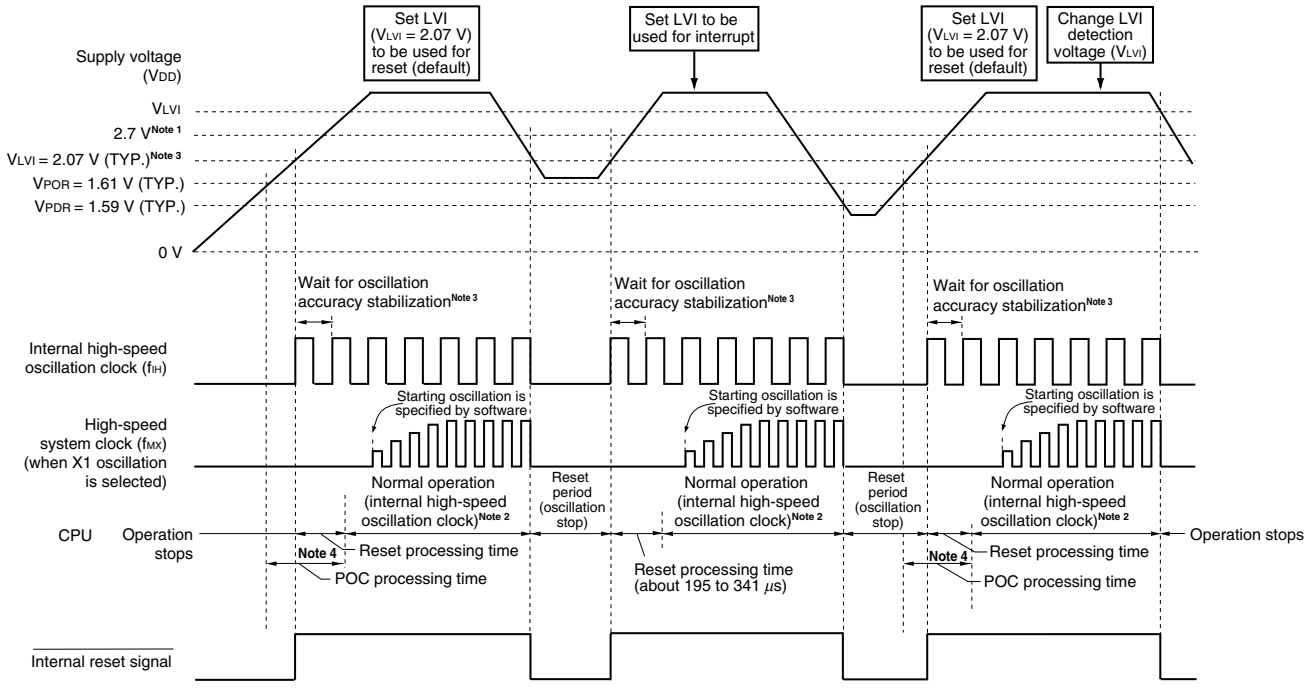
Cautions 1. Set the low-voltage detector by software after the reset status is released (see CHAPTER 22 LOW-VOLTAGE DETECTOR).

2. Some operations can also be executed while V_{DD} < 2.7 V (For details, figures of CHAPTER 29 ELECTRICAL SPECIFICATIONS AC Characteristics (1) Basic operation.

Remark V_{LVI}: LVI detection voltage
V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall 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 2.7 V ≤ V_{DD} ≤ 5.5 V. To make the state at lower than 2.7 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 (1.61 V (TYP.)) and starting normal operation.

- When the time to reach 2.07 V (TYP.) from 1.61 V (TYP.) is less than 5.8 ms:
  A POC processing time of about 2.12 to 5.84 ms is required between reaching 1.61 V (TYP.) and starting normal operation.
- When the time to reach 2.07 V (TYP.) from 1.61 V (TYP.) is greater than 5.8 ms:
  A reset processing time of about 195 to 341 μs is required between reaching 2.07 V (TYP.) and starting normal operation.

Cautions 1. Set the low-voltage detector by software after the reset status is released (see CHAPTER 22 LOW-VOLTAGE DETECTOR).

2. Some operations can also be executed while V_{DD} < 2.7 V (For details, figures of CHAPTER 29 ELECTRICAL SPECIFICATIONS AC Characteristics (1) Basic operation.

Remark V_{LVI}: LVI detection voltage
V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

21.4 Cautions for Power-on-Clear Circuit

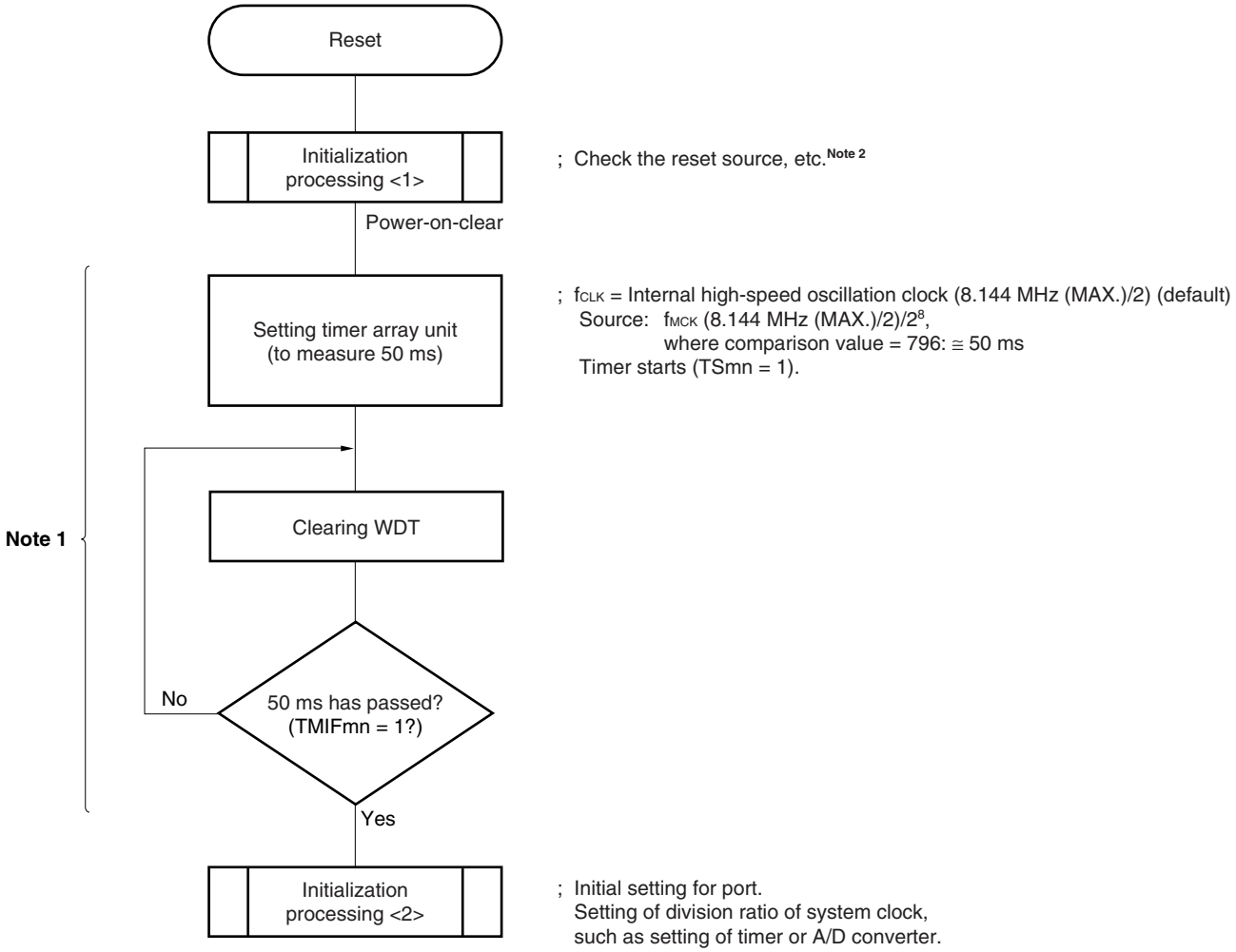
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POR}, V_{PDR}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller 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



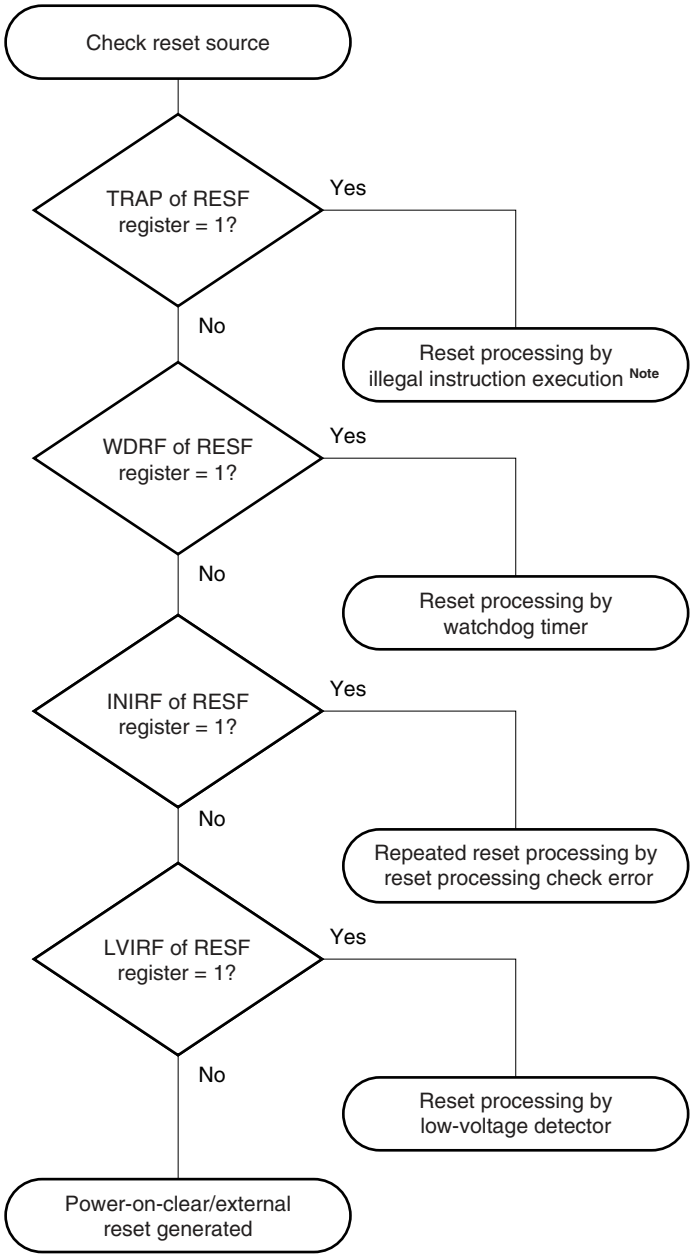
Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Remark n = 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 V ±0.1 V), and generates an internal reset 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 (V_{POR} = 1.61 V (TYP.)) or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.2 V). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 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},10 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

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 ≥ 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 ≥ V_{EXLVI}).

Remark LVISEL: Bit 2 of low-voltage detection register (LVIM)
LVIMD: Bit 1 of LVIM

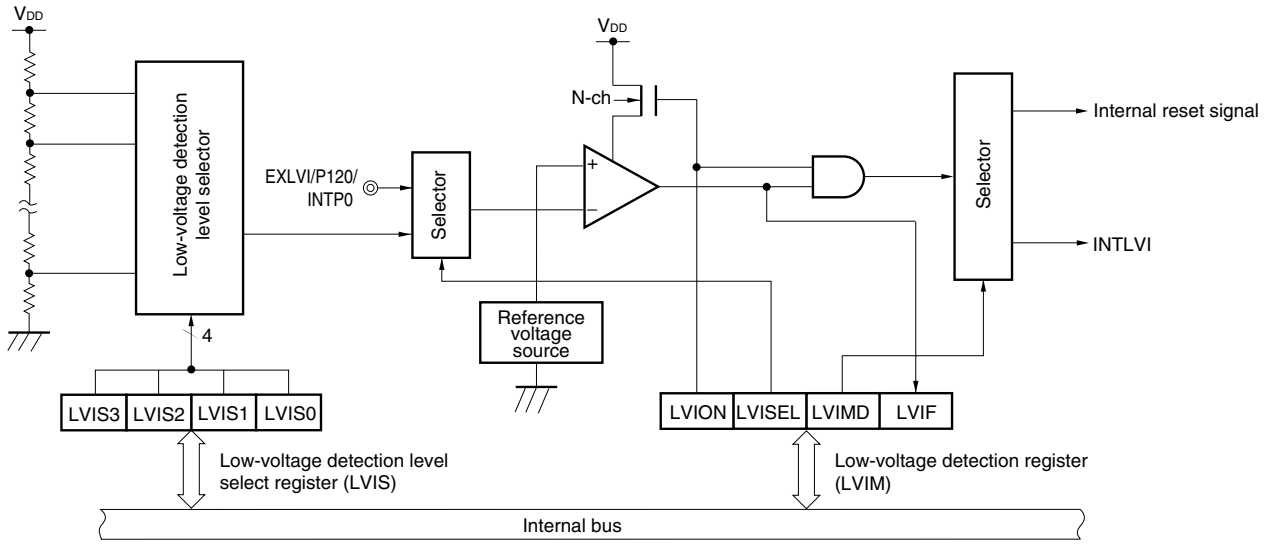
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

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[Note 1] R/W[Note 2]

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
LVIM	LVION	0	0	0	0	LVISEL	LVIMD	LVIF

LVION[Notes 3, 4]	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVISEL[Note 3]	Voltage detection selection
0	Detects level of supply voltage (V_{DD})
1	Detects level of input voltage from external input pin (EXLVI)

LVIMD[Note 3]	Low-voltage detection operation mode (interrupt/reset) selection
0	• LVISEL = 0: Generates an internal interrupt signal when the supply voltage (V_{DD}) drops lower than the detection voltage (V_{LVI}) ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$). • LVISEL = 1: Generates an interrupt signal when the input voltage from an external input pin (EXLVI) drops lower than the detection voltage (V_{EXLVI}) (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}).
1	• LVISEL = 0: Generates an internal reset signal when the supply voltage (V_{DD}) < detection voltage (V_{LVI}) and releases the reset signal when $V_{DD} \geq V_{LVI}$. • LVISEL = 1: Generates an internal reset signal when the input voltage from an external input pin (EXLVI) < detection voltage (V_{EXLVI}) and releases the reset signal when EXLVI ≥ V_{EXLVI}.

LVIF	Low-voltage detection flag
0	• LVISEL = 0: Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}), or when LVI operation is disabled • LVISEL = 1: Input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI}), or when LVI operation is disabled
1	• LVISEL = 0: Supply voltage (V_{DD}) < detection voltage (V_{LVI}) • LVISEL = 1: Input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI})

Notes 1. The reset value changes depending on the reset source and the setting of the option byte.
This register is not cleared (00H) by 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.

2. Bit 0 is read-only.

3. 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.

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 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))

The LVIF value for these periods may be set/cleared regardless of the voltage level, and can therefore not be used. Also, the LVIIF interrupt request flag may be set to 1 in these periods.

(Cautions are given on the next page.)

Cautions 1. To stop LVI, be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

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 LVIIF may be set to 1.

4. To read LVIM after writing this register, secure the time of one or more clock.

(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 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_{LVI0} (4.22 ±0.1 V)[Note]
0	0	0	1	V_{LVI1} (4.07 ±0.1 V)[Note]
0	0	1	0	V_{LVI2} (3.92 ±0.1 V)[Note]
0	0	1	1	V_{LVI3} (3.76 ±0.1 V)[Note]
0	1	0	0	V_{LVI4} (3.61 ±0.1 V)[Note]
0	1	0	1	V_{LVI5} (3.45 ±0.1 V)[Note]
0	1	1	0	V_{LVI6} (3.30 ±0.1 V)[Note]
0	1	1	1	V_{LVI7} (3.15 ±0.1 V)[Note]
1	0	0	0	V_{LVI8} (2.99 ±0.1 V)[Note]
1	0	0	1	V_{LVI9} (2.84 ±0.1 V)[Note]
Other than above				Setting prohibited

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 by LVI is effected.

Caution 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 an LVIIF flag with software because it 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 an LVIIF flag with software because it 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.

4. To read LVIM after writing this register, secure the time of one or more clock.

(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} = 1.21 V ±0.1 V), generates an internal reset signal when EXLVI < V_{EXLVI}, and releases internal reset when EXLVI ≥ 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 (V_{POR} = 1.61 V (TYP.)) or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.2 V). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 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 V ±0.1 V). When EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ 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 (V_{DD})

(a) When LVI default start function stopped is set (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 210 μs).

- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))

<6> Wait until it is checked that (supply voltage (V_{DD}) ≥ 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.

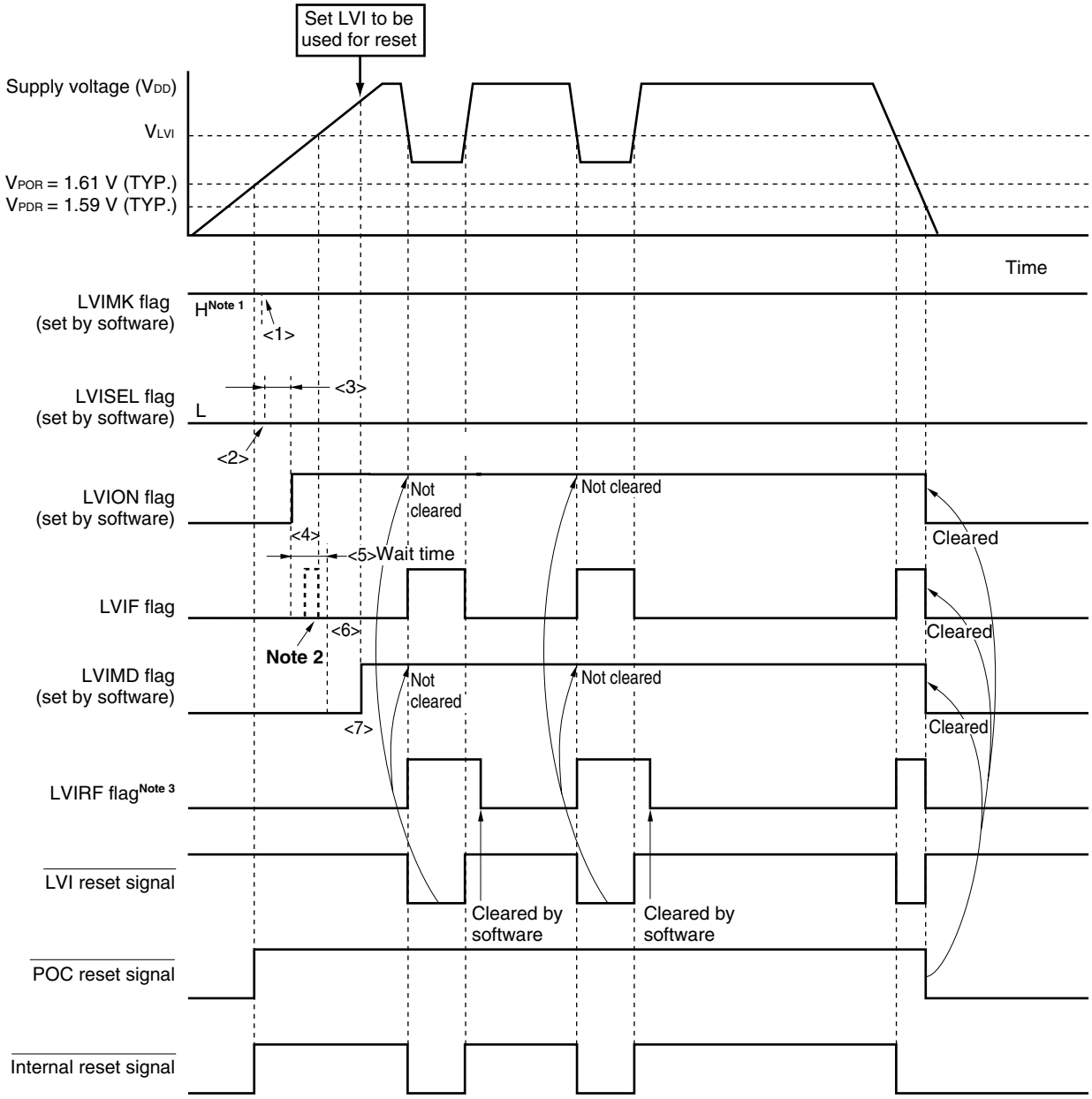
Cautions 1. Be sure to execute <1>. When LVIMK = 0, an interrupt may occur immediately after the processing in <4>.

2. If supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}) when LVIMD is set to 1, an internal reset signal is not generated.

- When stopping operation

Be sure to clear (0) LVIMD and then LVION by using a 1-bit memory manipulation instruction.

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 LVIIF 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**.

Remarks 1. <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 (LVIOFF = 1).**

2. V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(b) When LVI default start function enabled is set (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 V ±0.1 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}) ≥ detection voltage (V_{LVI})”)

  Figure 22-6 shows the timing of the internal reset signal generated by the low-voltage detector.

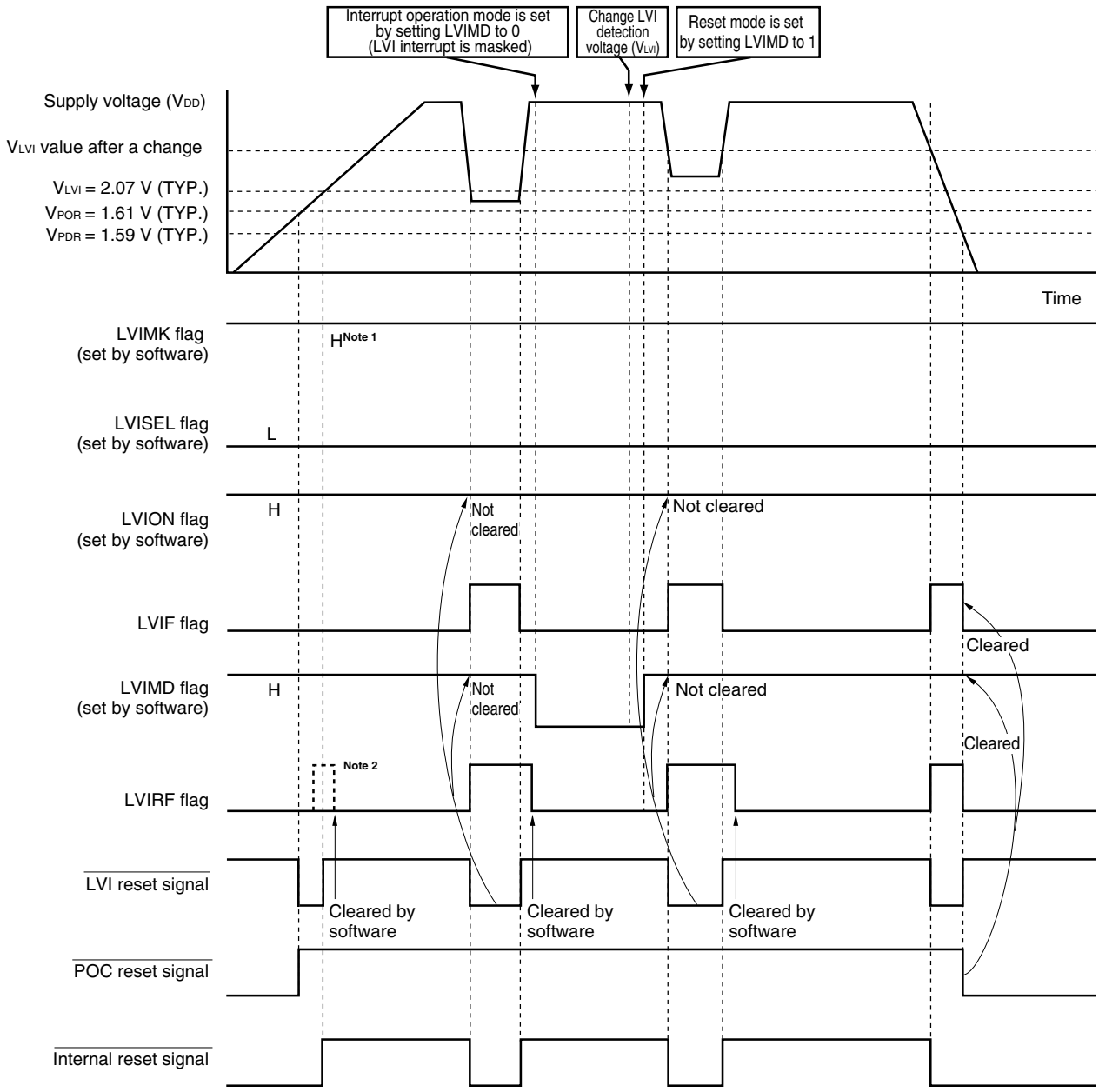
- When stopping operation

  Be sure to clear (0) LVIMD and then LVION by using a 1-bit memory manipulation instruction.

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 μs 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**.

Remark V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(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 210 μs).

- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))

<5> Wait until it is checked that (input voltage from external input pin (EXLVI) ≥ 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. Be sure to execute <1>. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.

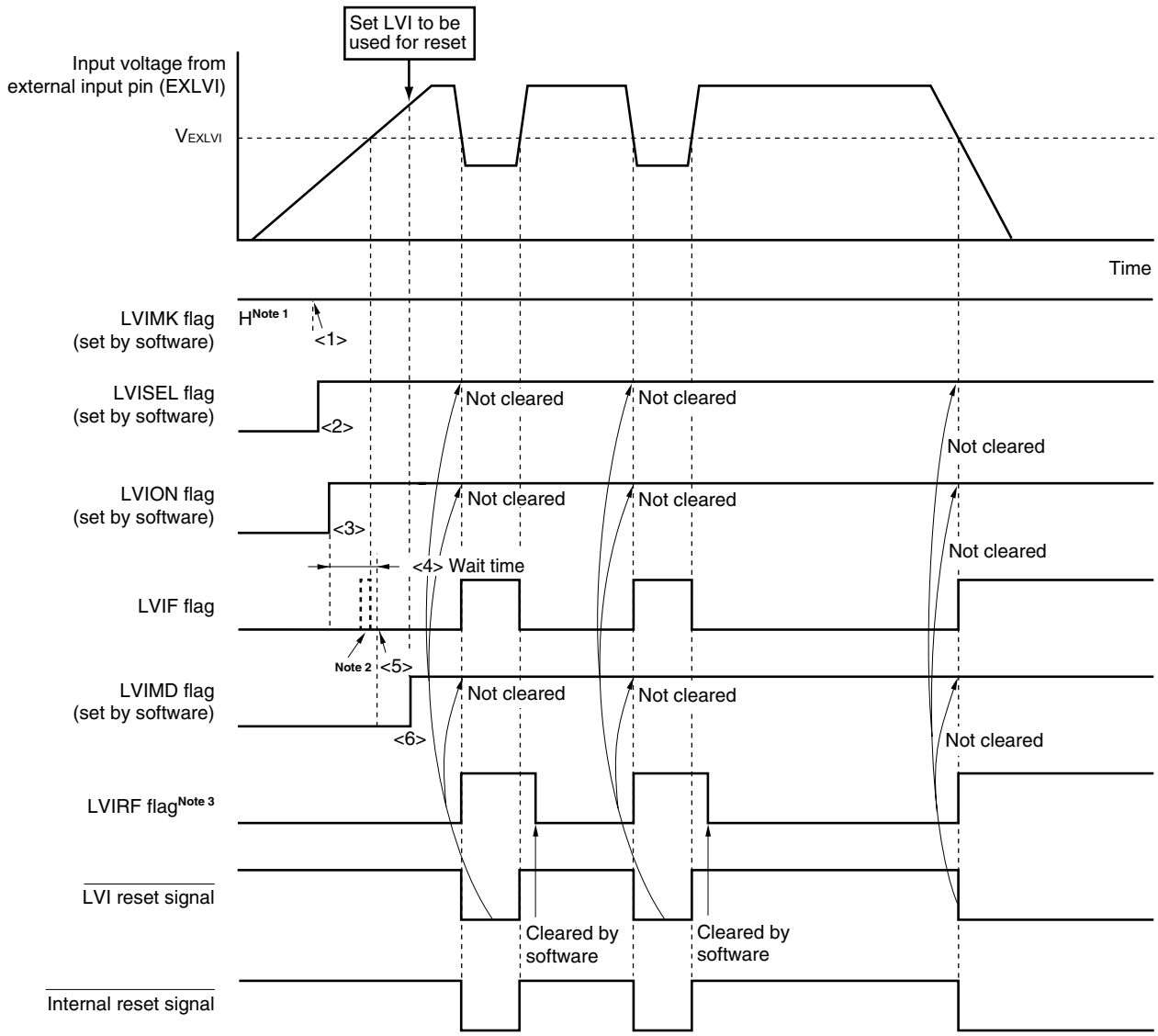
2. If input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.

3. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation

Be sure to clear (0) LVIMD and then LVION by using a 1-bit memory manipulation instruction.

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 LVIIF 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 (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 210 μs).
- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))

<6> Confirm that "supply voltage (V_{DD}) ≥ 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).

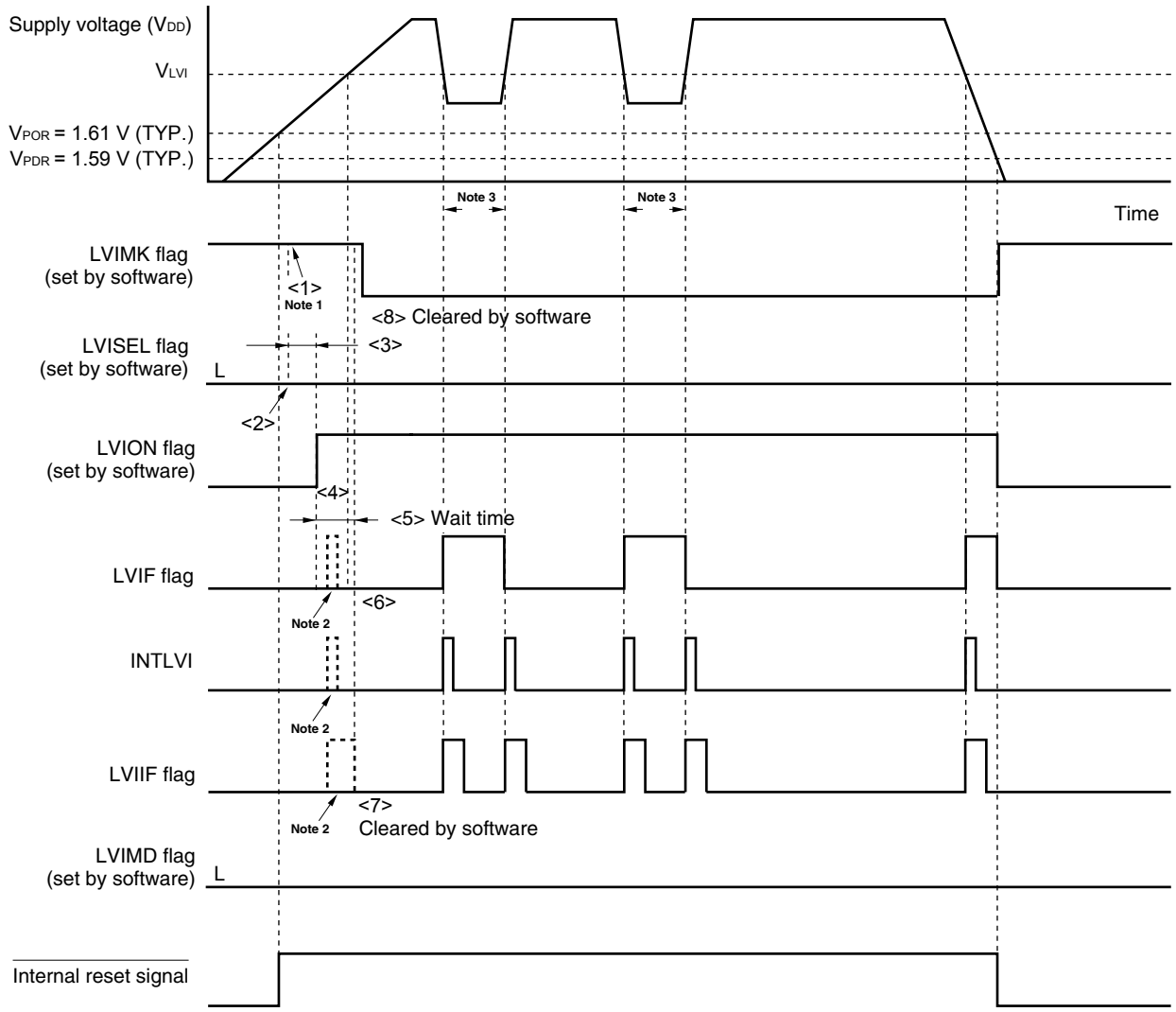
<9> Execute the EI instruction (when vector interrupts are used).

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

Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

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_{DD}) is less than or equal to the detection voltage (V_{LVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

Remarks 1. <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 (LVIOFF = 1).**

2. V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(b) When LVI default start function enabled is set (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 V ±0.1 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 from the state of "Supply voltage (V_{DD}) ≥ 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

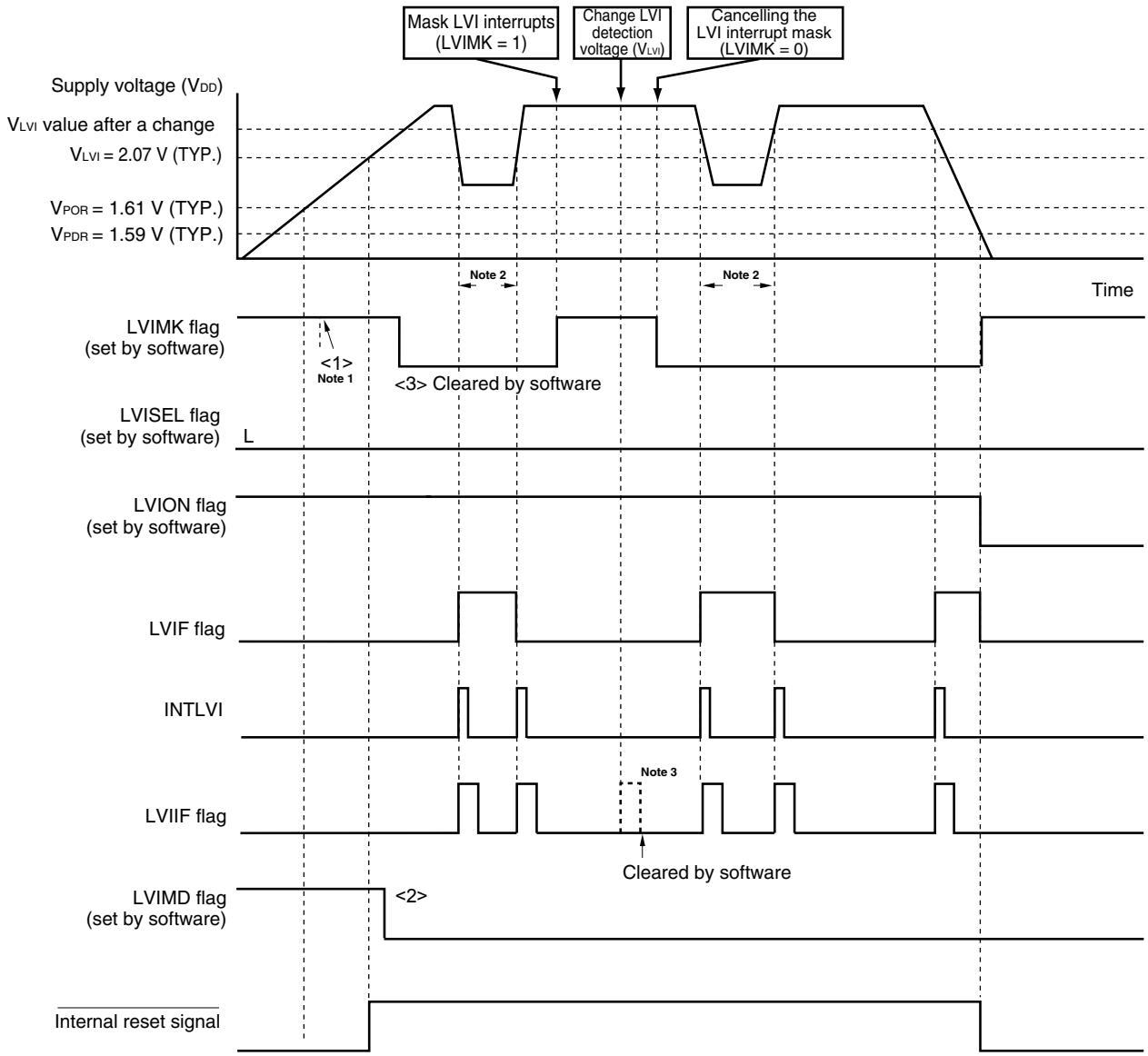
Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

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 μs 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 LVIIF may be set to 1.

3. The LVIIF flag may be set when the LVI detection voltage is changed.

Remarks 1. <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 (LVIOFF = 0).**

2. V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(2) When detecting level of input voltage from external input pin (EXLVI)

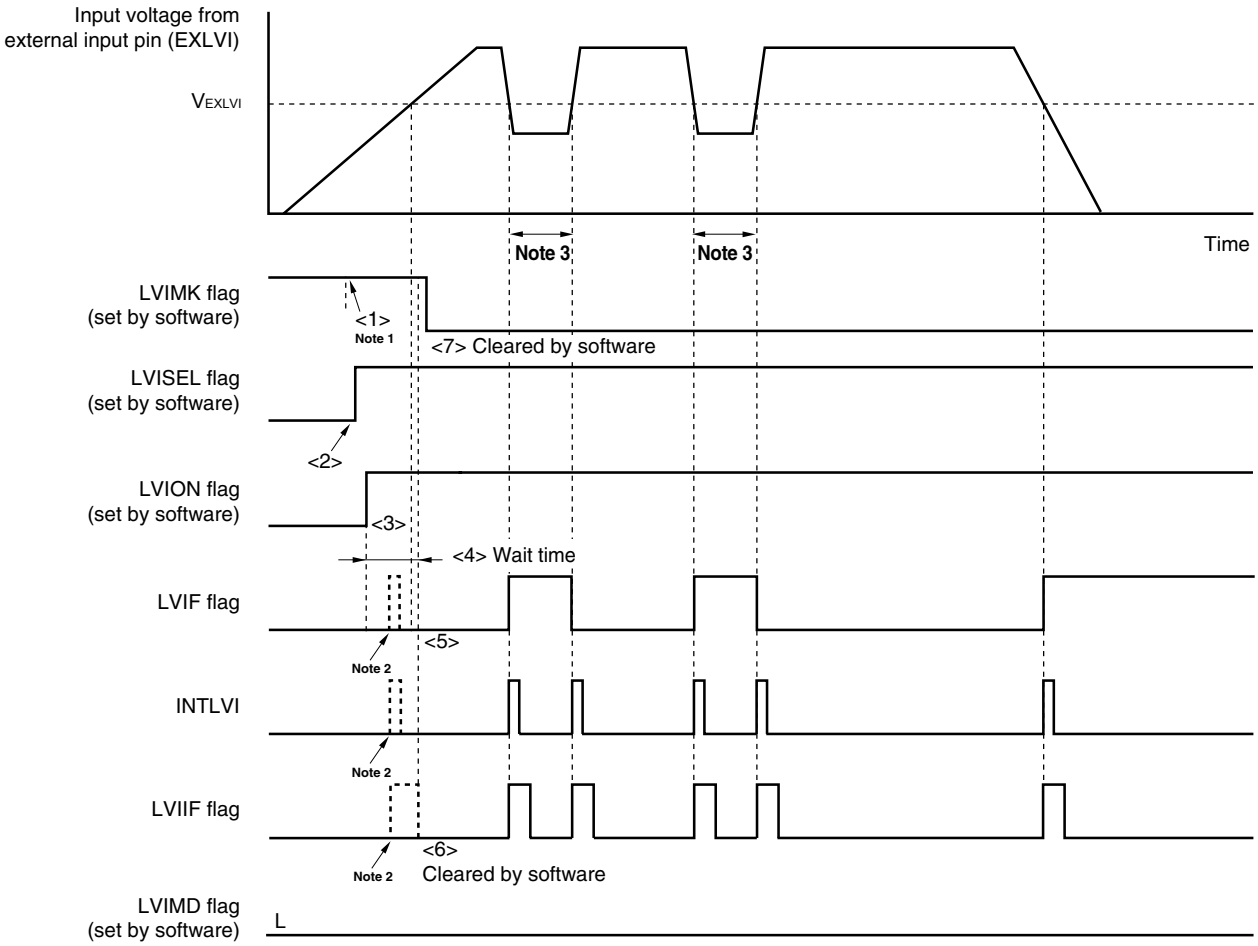
- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
    Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (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 210 μs).
    - Operation stabilization time (10 μs (MAX.))
    - Minimum pulse width (200 μs (MIN.))
  - <5> Confirm that "input voltage from external input pin (EXLVI) ≥ 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 (LVIIF) 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
  Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

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>

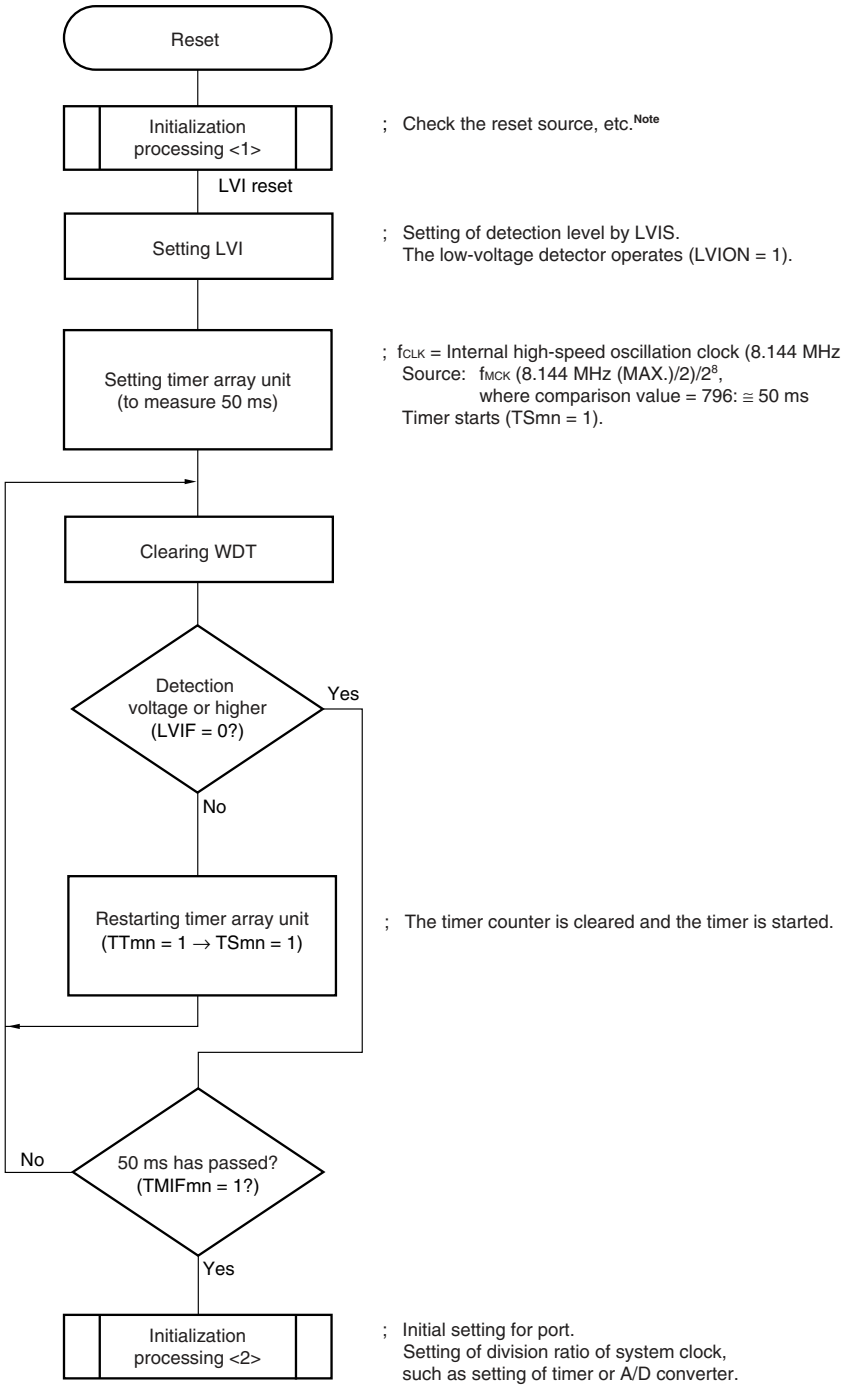
After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (see **Figure 22-11**).

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.

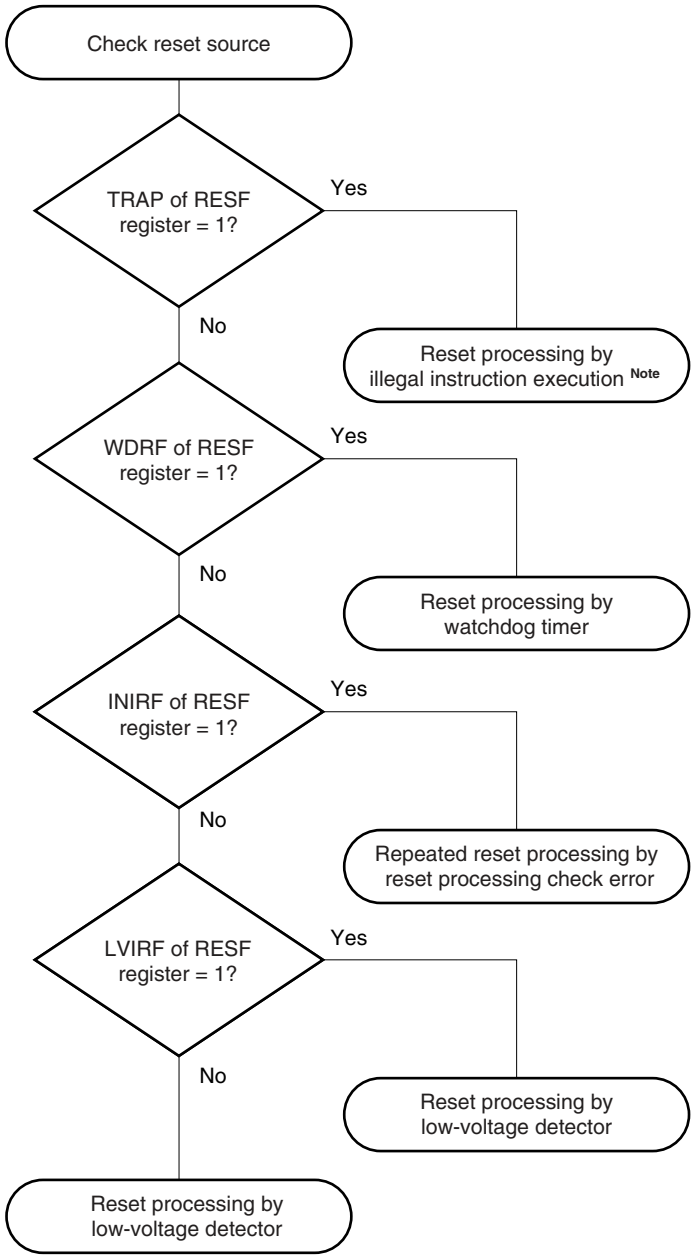
Remarks 1. If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

2. m = 0, 1, n = 0 to 7, mn = 00 to 07, 10 to 12

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 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}) ≥ 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.

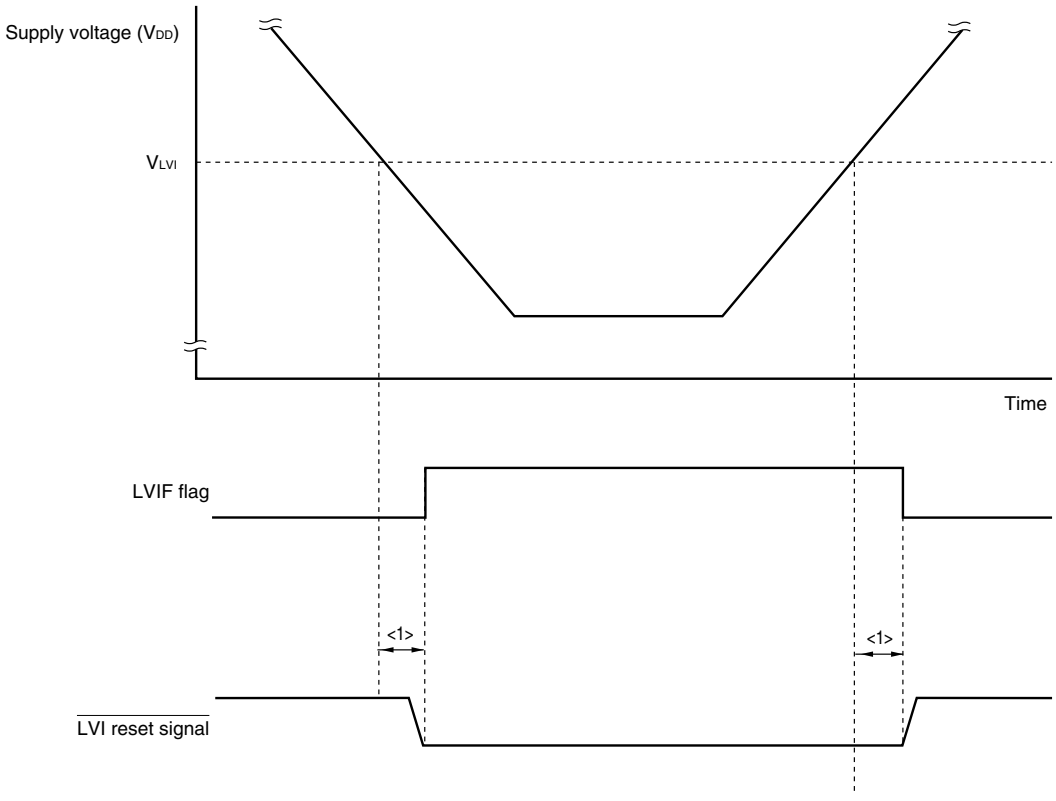
- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

(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}) ≤ supply voltage (V_{DD}) until the time LVI reset has been released (see **Figure 22-12**).

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 μs (MIN.))

CHAPTER 23 REGULATOR

23.1 Regulator Overview

The 78K0R/Kx3-C contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). 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.

The regulator output voltage is normally 2.4 V (typ.), and in the low power consumption 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 power consumption mode (1.8 V)
00H	Switches normal power consumption mode (2.4 V) and low power consumption mode (1.8 V) according to the condition (refer to **Table 23-1**)
Other than above	Setting prohibited

Cautions 1. When using the setting fixed to the low power consumption 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 $f_{CLK} \leq 1$ MHz

<When the internal high-speed oscillation clock, external input clock, or subsystem clock are selected for the CPU clock> $f_{CLK} \leq 1$ MHz

(Caution is given on the next page.)

Cautions 2. A wait is required to change the operation speed mode control register (OSMC) after changing the RMC register. Wait for 3.5 ms by software when setting to low power consumption mode and 10 μs when setting to normal power consumption mode mode, as described in the procedure shown below.

- **When setting to low power consumption mode**
  - **<1> Select a frequency of 1 MHz for f_{CLK}.**
  - **<2> Set RMC to 5AH (set the regulator to low power consumption mode).**
  - **<3> Wait for 3.5 ms.**
  - **<4> Set FLPC and FSEL of OSMC to 1 and 0, respectively.**

- **When setting to normal power consumption mode**
  - **<1> Set RMC to 00H (set the regulator to normal power consumption mode).**
  - **<2> Wait for 10 μs.**
  - **<3> Change FLPC and FSEL of OSMC.**
  - **<4> Change the f_{CLK} frequency.**

Table 23-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
Low power consumption mode	1.8 V	In STOP mode (except during OCD mode)
		When both the high-speed system clock (f_{MX}), the internal high-speed oscillation clock (f_{IH}), and the 20 MHz internal high-speed oscillation clock (f_{IH20}) are stopped during CPU operation with the subsystem clock (f_{XT})
		When both the high-speed system clock (f_{MX}), the internal high-speed oscillation clock (f_{IH}), and the 20 MHz internal high-speed oscillation clock (f_{IH20}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{XT}) has been set
Normal power consumption mode	2.4 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/Kx3-C 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).
- ○ Setting of internal high-speed oscillator frequency
  - Select from 8 MHz or 20 MHz.

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[Note 1]

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period[Note 2]
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 (f_{IL} = 33 kHz (MAX.))
0	0	0	$2^7/f_{IL}$ (3.88 ms)
0	0	1	$2^8/f_{IL}$ (7.76 ms)
0	1	0	$2^9/f_{IL}$ (15.52 ms)
0	1	1	$2^{10}/f_{IL}$ (31.03 ms)
1	0	0	$2^{12}/f_{IL}$ (124.12 ms)
1	0	1	$2^{14}/f_{IL}$ (496.48 ms)
1	1	0	$2^{15}/f_{IL}$ (992.97 ms)
1	1	1	$2^{17}/f_{IL}$ (3971.88 ms)

Figure 24-1. Format of User Option Byte (000C0H/010C0H) (2/2)

Address: 000C0H/010C0H[Note 1]

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[Note 2]
1	Counter operation enabled in HALT/STOP mode

Notes 1. Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

2. The window open period is 100% when WDSTBYON = 0, regardless the value of 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.

Remark f_{IL}: Internal low-speed oscillation clock frequency

Figure 24-2. Format of User Option Byte (000C1H/010C1H)

Address: 000C1H/010C1H[Note 1]

7	6	5	4	3	2	1	0
1	1	1	1	1	FRQSEL2	FRQSEL1	LVIOFF

FRQSEL2	FRQSEL1	Internal high-speed oscillator frequency
0	1	8 MHz/20 MHz [Note 2]
Other than the above		Setting prohibited

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)

Notes 1. Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

2. When 8 MHz or 20 MHz has been selected, the 8 MHz internal high-speed oscillator automatically starts oscillating after reset release. To use the 20 MHz internal high-speed oscillator to operate the microcontroller, oscillation is started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1.

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

Cautions 1. Be sure to set bits 7 to 3 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 μs 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[Note]

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[Note]

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Enabling on-chip debug operation. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enabling on-chip debug operation. 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/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
      DB          0FBH    ; Select 8 MHz or 20 MHz for internal high-speed oscillator
                          ; 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/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
      DB          0FBH    ; Select 8 MHz or 20 MHz for internal high-speed oscillator
                          ; 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/Kx3-C 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/Kx3-C.

- 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/Kx3-C 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/Kx3-C is mounted on the target system.

Remark The FA series are products of Naito Densei Machida Mfg. Co., Ltd.

Table 25-1. Wiring Between 78K0R/Kx3-C and Dedicated Flash Memory Programmer

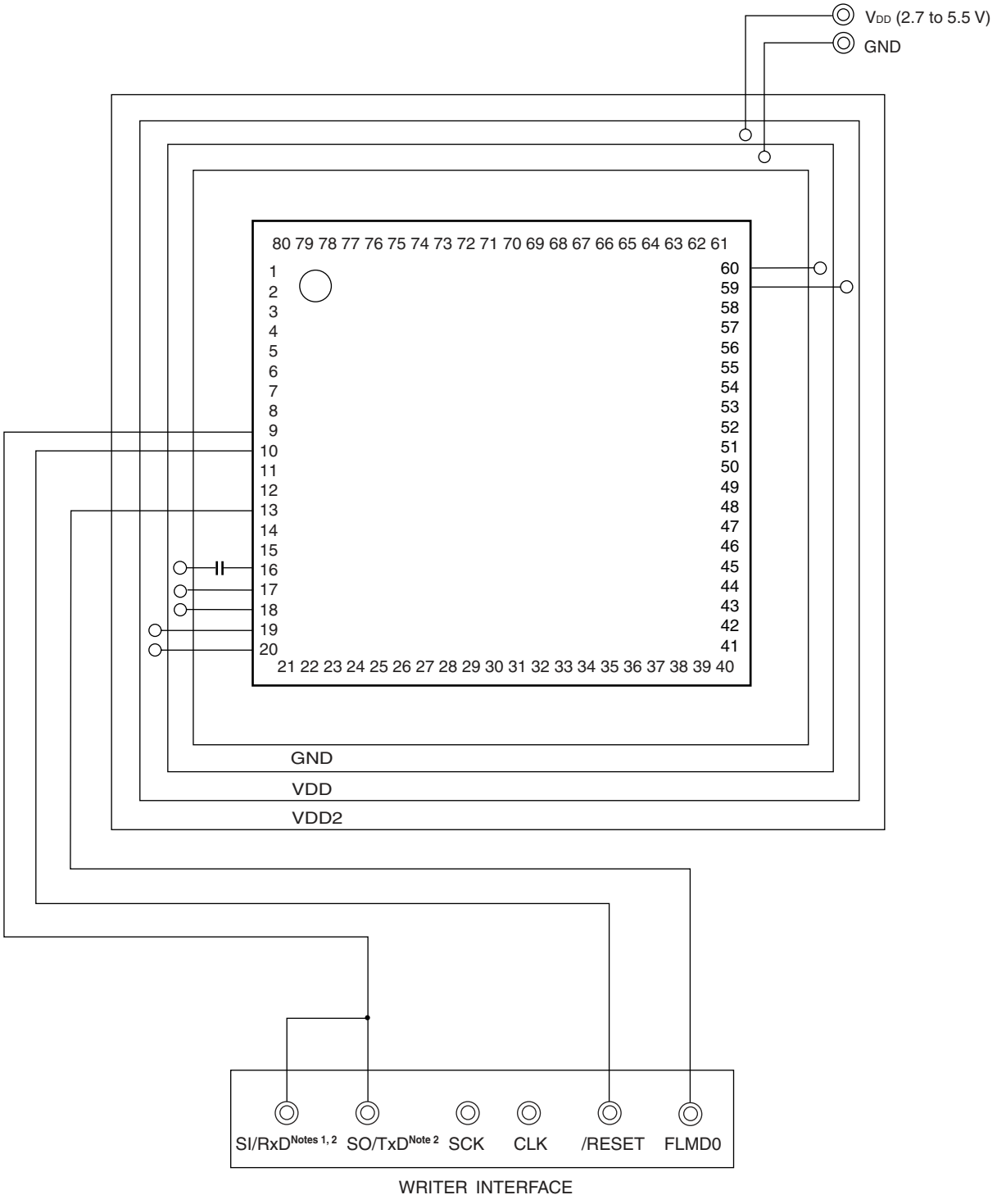
Pin Configuration of Dedicated Flash Memory Programmer			78K0R/KF3-C		78K0R/KG3-C	
Signal Name	I/O	Pin Function	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD[Notes 1, 2]	Input	Receive signal	TOOL0/P40	9	TOOL0/P40	12
SO/TxD[Note 2]	Output	Transmit signal				
SCK	Output	Transfer clock	–	–	–	–
CLK	Output	Clock output	–	–	–	–
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	10	$\overline{\text{RESET}}$	13
FLMD0	Output	Mode signal	FLMD0	13	FLMD0	16
V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	19	V_{DD}	22
			EV_{DD}	20	EV_{DD0}	23
					EV_{DD1}	53
			AV_{REF}	59	AV_{REF}	73
GND	–	Ground	V_{SS}	17	V_{SS}	20
			EV_{SS}	18	EV_{SS0}	21
					EV_{SS1}	43
			AV_{SS}	60	AV_{SS}	74

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 connection when using the adapter for flash writing are shown below.

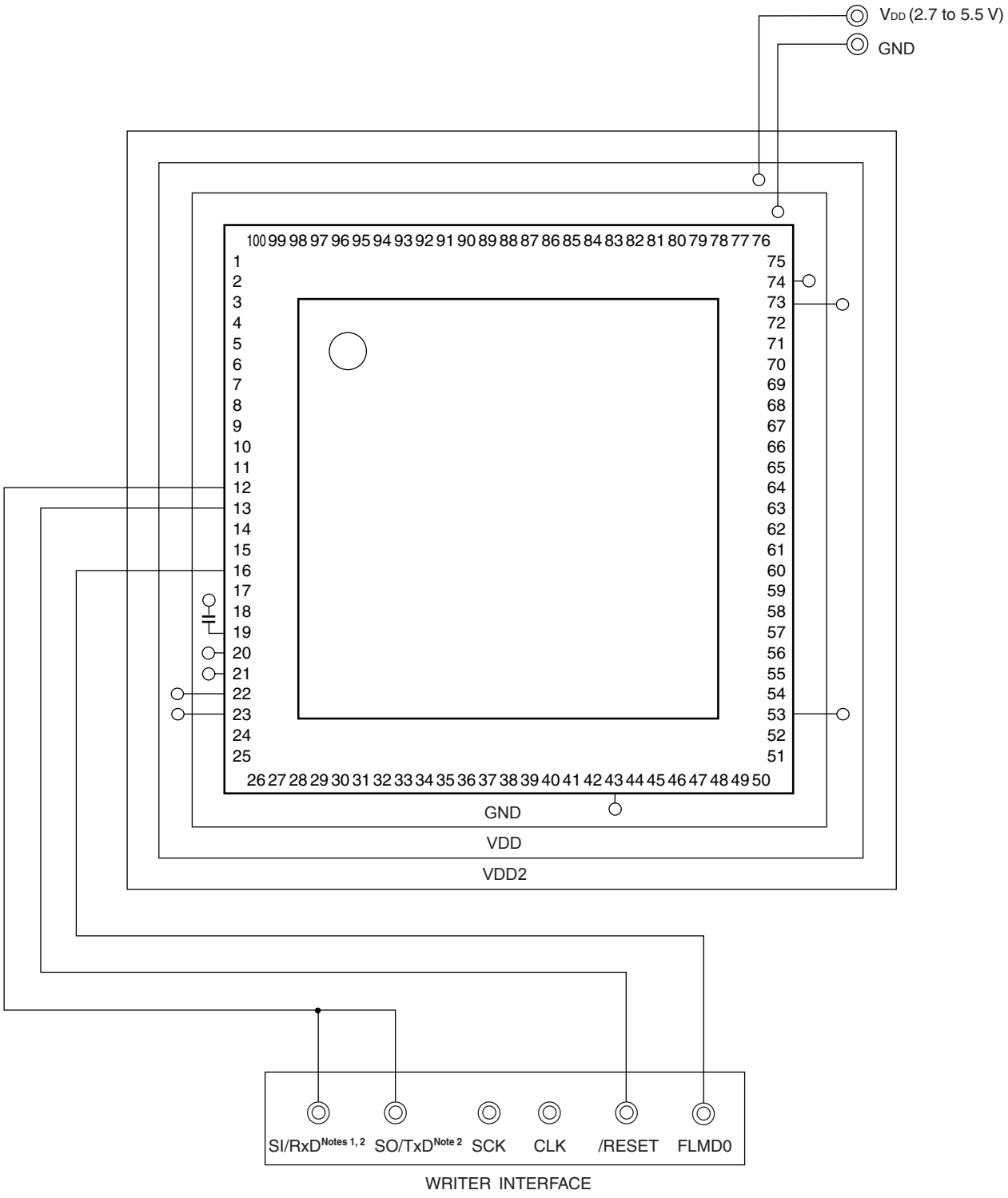
Figure 25-1. Example of Wiring Adapter for Flash Memory Writing (78K0R/KF3-C)



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.

Figure 25-2. Example of Wiring Adapter for Flash Memory Writing (78K0R/KG3-C)



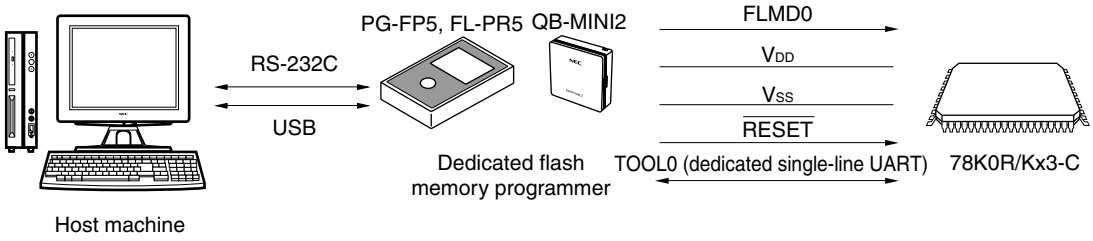
Notes 1. This pin is not required to be connected when using PG-FP5 or FL-PR5.

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/Kx3-C is illustrated below.

Figure 25-3. Environment for Writing Program to Flash Memory



Remark FL-PR5 is product of Naito Densei Machida Mfg. Co., Ltd.

A host machine that controls the dedicated flash memory programmer is necessary.

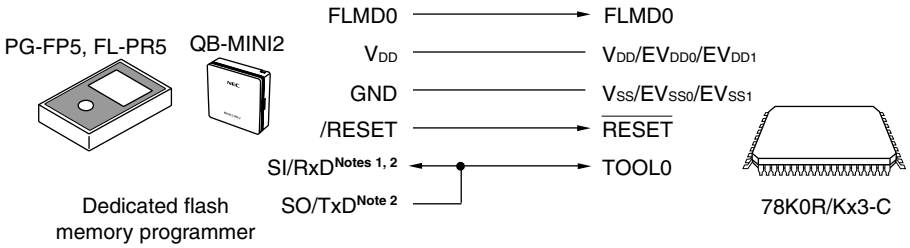
To interface between the dedicated flash memory programmer and the 78K0R/Kx3-C, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

25.3 Communication Mode

Communication between the dedicated flash memory programmer and the 78K0R/Kx3-C is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the 78K0R/Kx3-C.

Transfer rate: 115,200 bps, 250,000 bps, 500,000 bps, 1 Mbps

Figure 25-4. 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.

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

The dedicated flash memory programmer generates the following signals for the 78K0R/Kx3-C. See the manual of PG-FP5, FL-PR5, or MINICUBE2 for details.

Table 25-2. Pin Connection

Dedicated Flash Memory Programmer			78K0R/KF3-C	78K0R/KG3-C	Connection
Signal Name	I/O	Pin Function	Pin Name	Pin Name	
FLMD0	Output	Mode signal	FLMD0	FLMD0	◎
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, EV_{DD}, AV_{REF}	V_{DD}, EV_{DD0}, EV_{DD1}, AV_{REF}	◎
GND	–	Ground	V_{SS}, EV_{SS}, AV_{SS}	V_{SS}, EV_{SS0}, EV_{SS1}, AV_{SS}	◎
CLK	Output	Clock output	–	–	×
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$	◎
SI/RxD Notes 1, 2	Input	Receive signal	TOOL0	TOOL0	◎
SO/TxD Note 2	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.

Remarks 1. ◎: Be sure to connect the pin.

×: The pin does not have to be connected.

2. For the pins not to be connected the dedicated flash memory programmer, it is recommended to perform the processing described under the “Recommended Connection of Unused Pins” shown in **Table 2-3. Connection of Unused Pins (78K0R/KF3-C)** or **Table 2-3. Connection of Unused Pins (78K0R/KG3-C)**.

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_{DD} 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_{SS} 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.

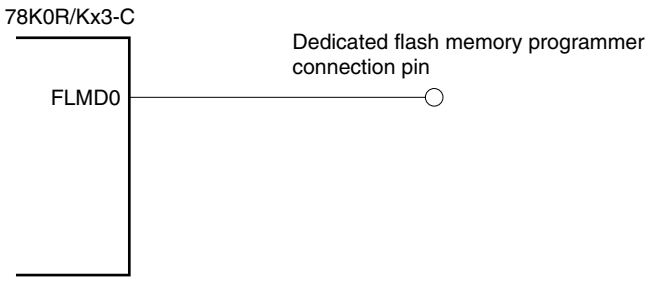
Self programming and the rewriting of flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V_{SS} pin.

(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 kΩ to 200 kΩ.

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

Figure 25-5. 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_{DD0} or EV_{DD1} via an external resistor.

When on-chip debugging is enabled in the normal operation mode, pull this pin up by connecting it to EV_{DD0} or EV_{DD1} via an external resistor, and be sure to keep inputting the V_{DD} level to the TOOL0 pin before reset is released (pulling down this pin is prohibited).

Remarks 1. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

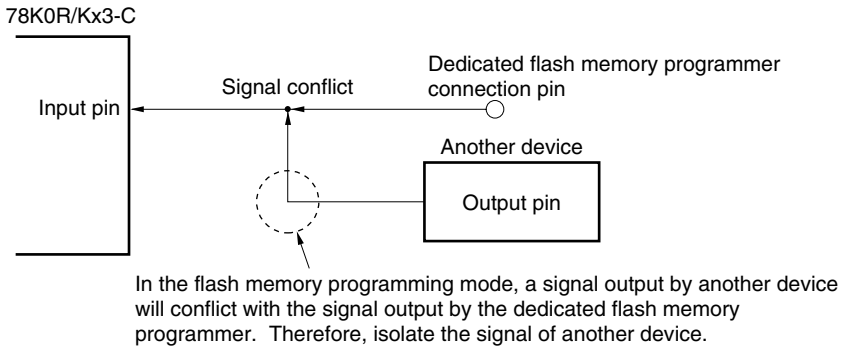
2. The SAU and IICA pins are not used for communication between the 78K0R/Kx3-C and dedicated flash memory programmer, because single-line UART is used.

25.4.3 $\overline{\text{RESET}}$ 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 25-6. Signal Conflict ($\overline{\text{RESET}}$ Pin)



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 μF) in the same manner as during normal operation. 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_{REF}, and AV_{SS}) as those in the normal operation mode.

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

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 kΩ 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 25-7. 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

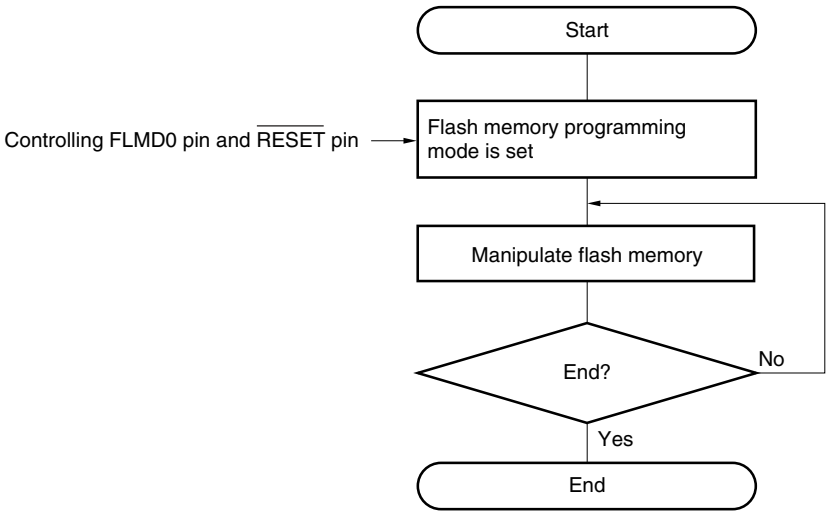
FLMDPUP	Software control of FLMD0 pin
0	Selects pull-down
1	Selects pull-up

25.6 Programming Method

25.6.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 25-8. 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/Kx3-C 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 25-9. Flash Memory Programming Mode

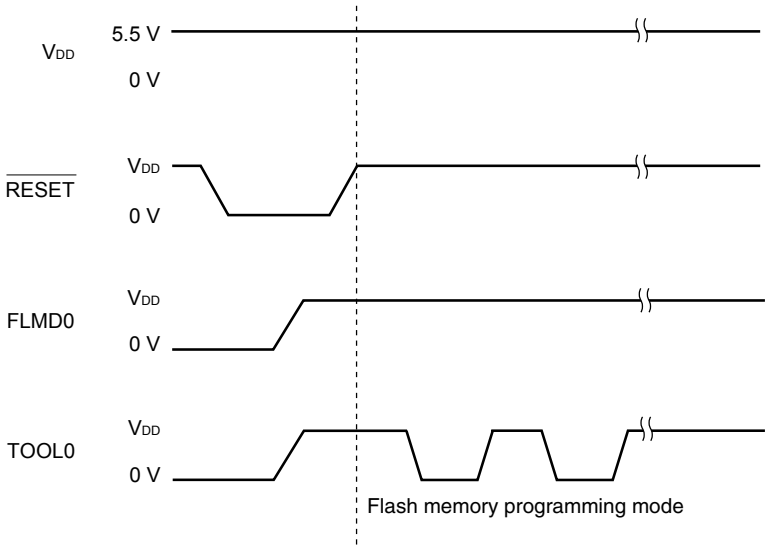


Table 25-3. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0 V	Normal operation mode
V_{DD}	Flash memory programming mode

25.6.3 Selecting communication mode

Communication mode of the 78K0R/Kx3-C is as follows.

Table 25-4. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed[Note 2]	Frequency	Multiply Rate	
1-line mode (single-line UART)	UART	115,200 bps, 250,000 bps, 500,000 bps, 1 Mbps	–	–	TOOL0

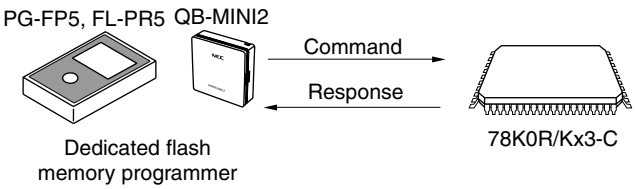
Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

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/Kx3-C communicates with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0R/Kx3-C are called commands, and the signals sent from the 78K0R/Kx3-C to the dedicated flash memory programmer are called response.

Figure 25-10. Communication Commands



The flash memory control commands of the 78K0R/Kx3-C are listed in the table below. All these commands are issued from the programmer and the 78K0R/Kx3-C 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/Kx3-C information (such as the part number and flash memory configuration).
	Version Get	Gets the 78K0R/Kx3-C 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/Kx3-C returns a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0R/Kx3-C 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/Kx3-C 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 block erase command and write command on boot cluster 0 (00000H to 00FFFH) in the flash memory is prohibited by this setting. In addition, execution of the batch erase (chip erase) command.

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/Kx3-C security function is enabled.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **25.8.2** for details).

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[Note].
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 shield window function (see **25.8.2** for details).

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 Flash Memory Programming by Self-Programming

The 78K0R/Kx3-C 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/Kx3-C 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.

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.

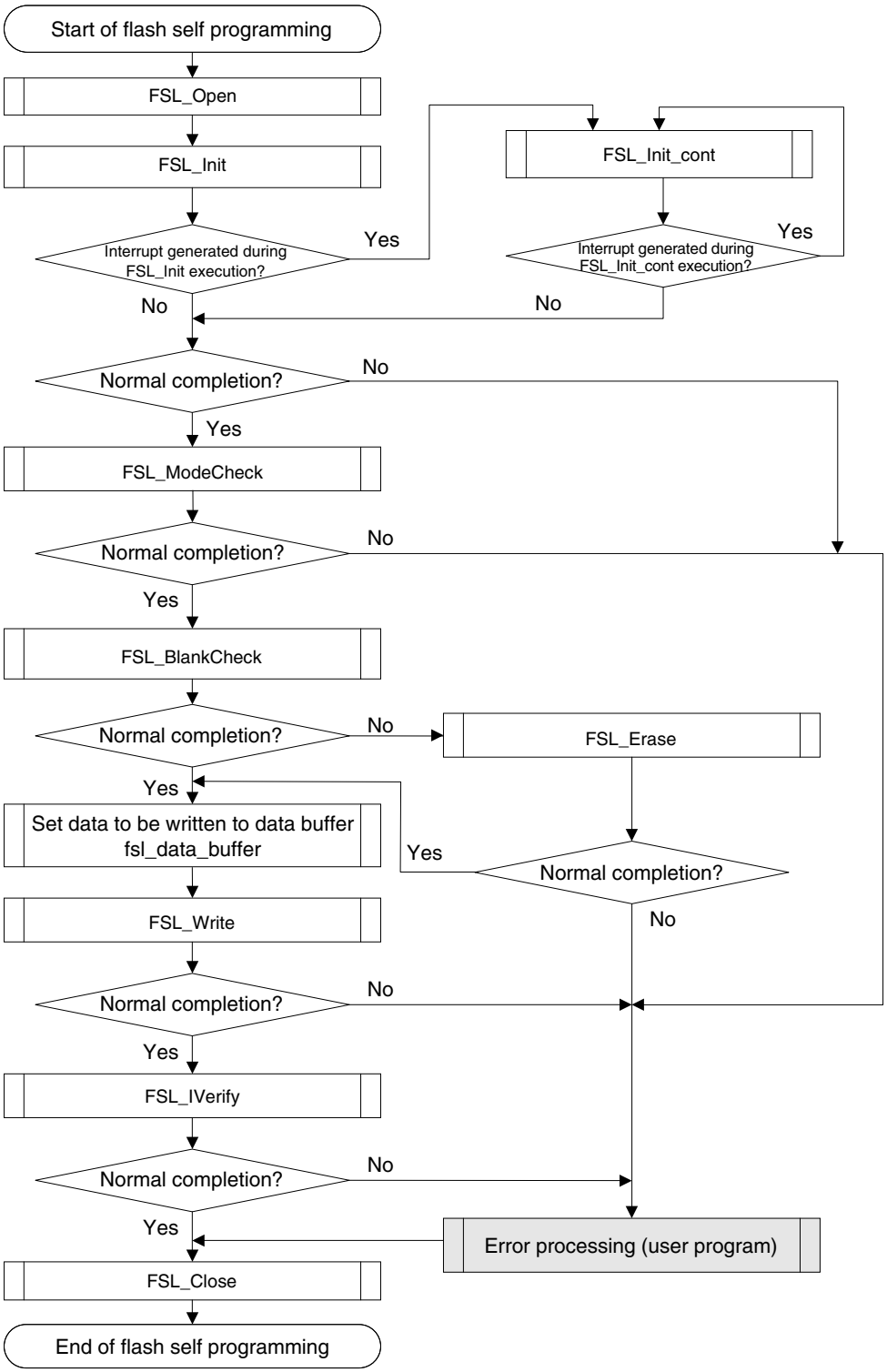
4. Disable DMA operation (DENn = 0) during the execution of self programming library functions.

Remarks 1. For details of the self-programming function and the 78K0R/Kx3-L self-programming library, refer to **78K0R Microcontroller Self Programming Library Type2 User's Manual (U19193E).**

2. For details of the time required to execute self programming, see the notes on use that accompany the flash self programming library tool.

The following figure illustrates a flow of rewriting the flash memory by using a self programming library.

Figure 25-11. Flow of Self Programming (Rewriting Flash Memory)



25.8.1 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.

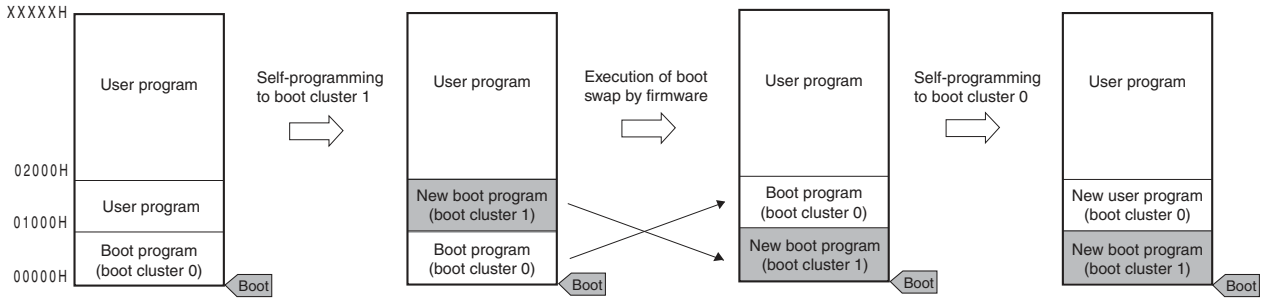
The boot swap function is used to avoid this problem.

Before erasing boot cluster 0[Note], 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/Kx3-C, 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 25-12. Boot Swap Function

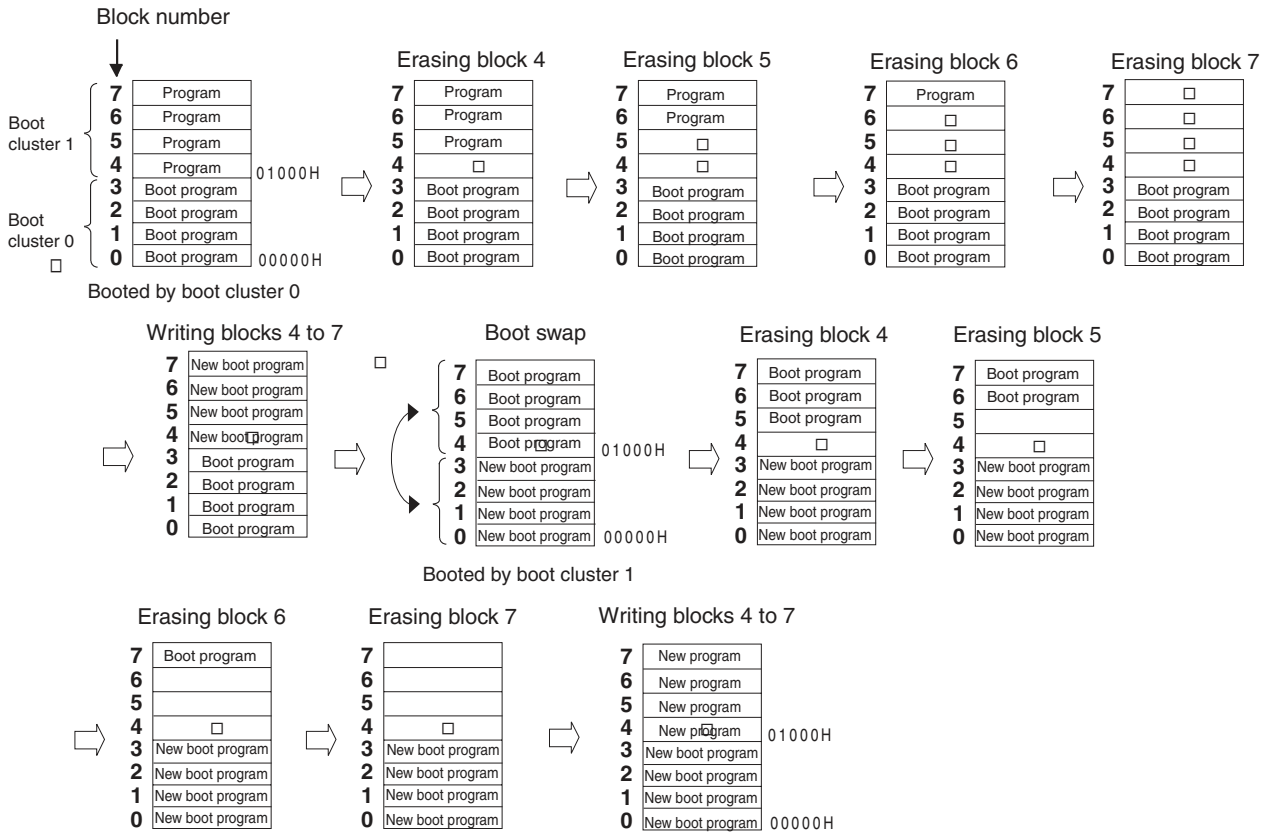


In an example of above figure, it is as follows.

Boot cluster 0: Boot program area before boot swap

Boot cluster 1: Boot program area after boot swap

Figure 25-13. Example of Executing Boot Swapping



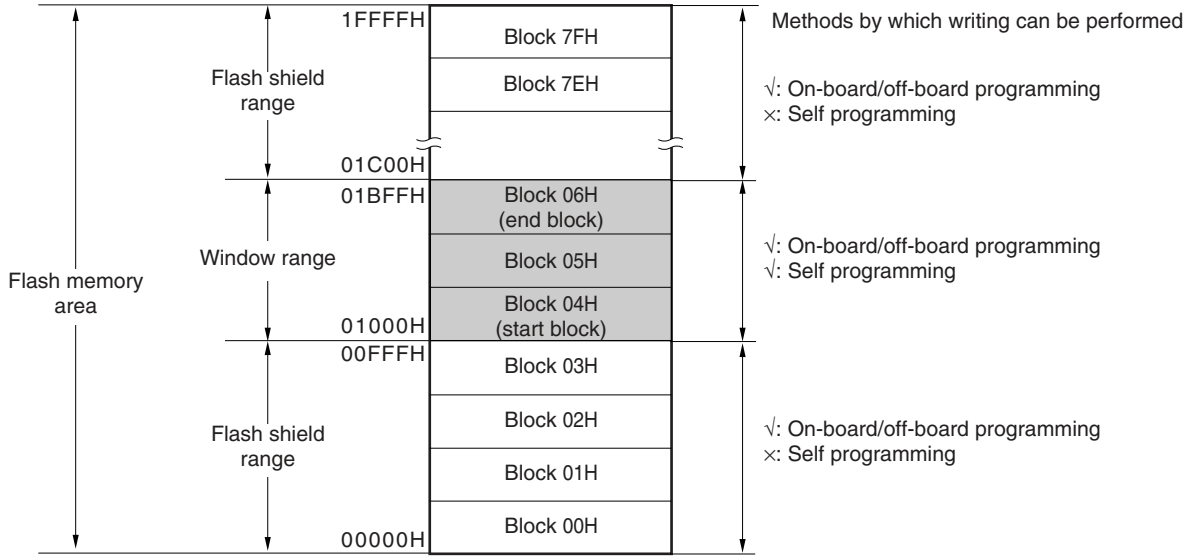
25.8.2 Flash shield window function

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 25-14. Flash Shield Window Setting Example
(Target Devices: μPD78F1847A, 78F1849A 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 25-9. 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 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.

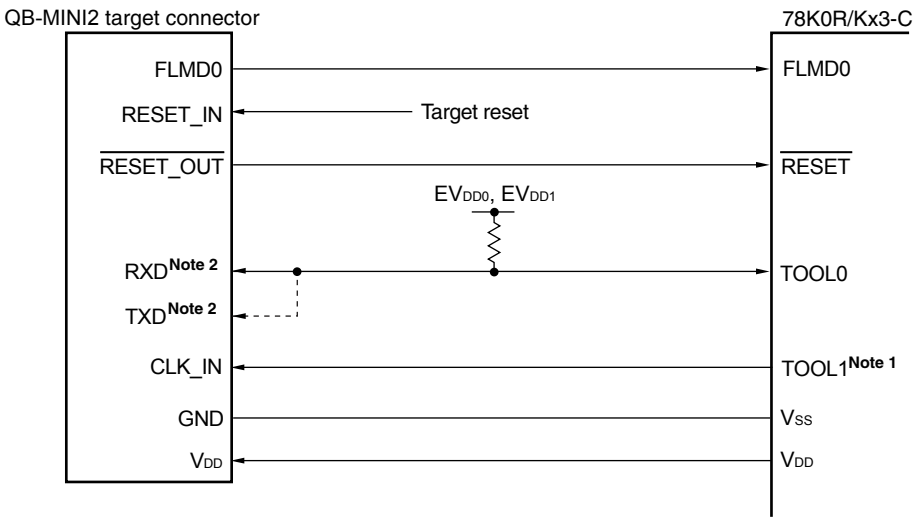
CHAPTER 26 ON-CHIP DEBUG FUNCTION

26.1 Connecting QB-MINI2 to 78K0R/Kx3-C

The 78K0R/Kx3-C uses the V_{DD}, FLMD0, $\overline{RESET}$, TOOL0, TOOL1[Note 1], and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2).

Caution The 78K0R/Kx3-C has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 26-1. Connection Example of QB-MINI2 and 78K0R/Kx3-C



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 2-3 Connection of Unused Pins (78K0R/KF3-C)** and **Table 2-4 Connection of Unused Pins (78K0R/KG3-C)** 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-MIN2. 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.

Remarks 1. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

2. The FLMD0 pin is recommended to be open for self-programming in on-chip debugging. To pull down externally, use a resistor of 100 kΩ 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
1-line mode	Available
2-line mode	None

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/Kx3-C 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

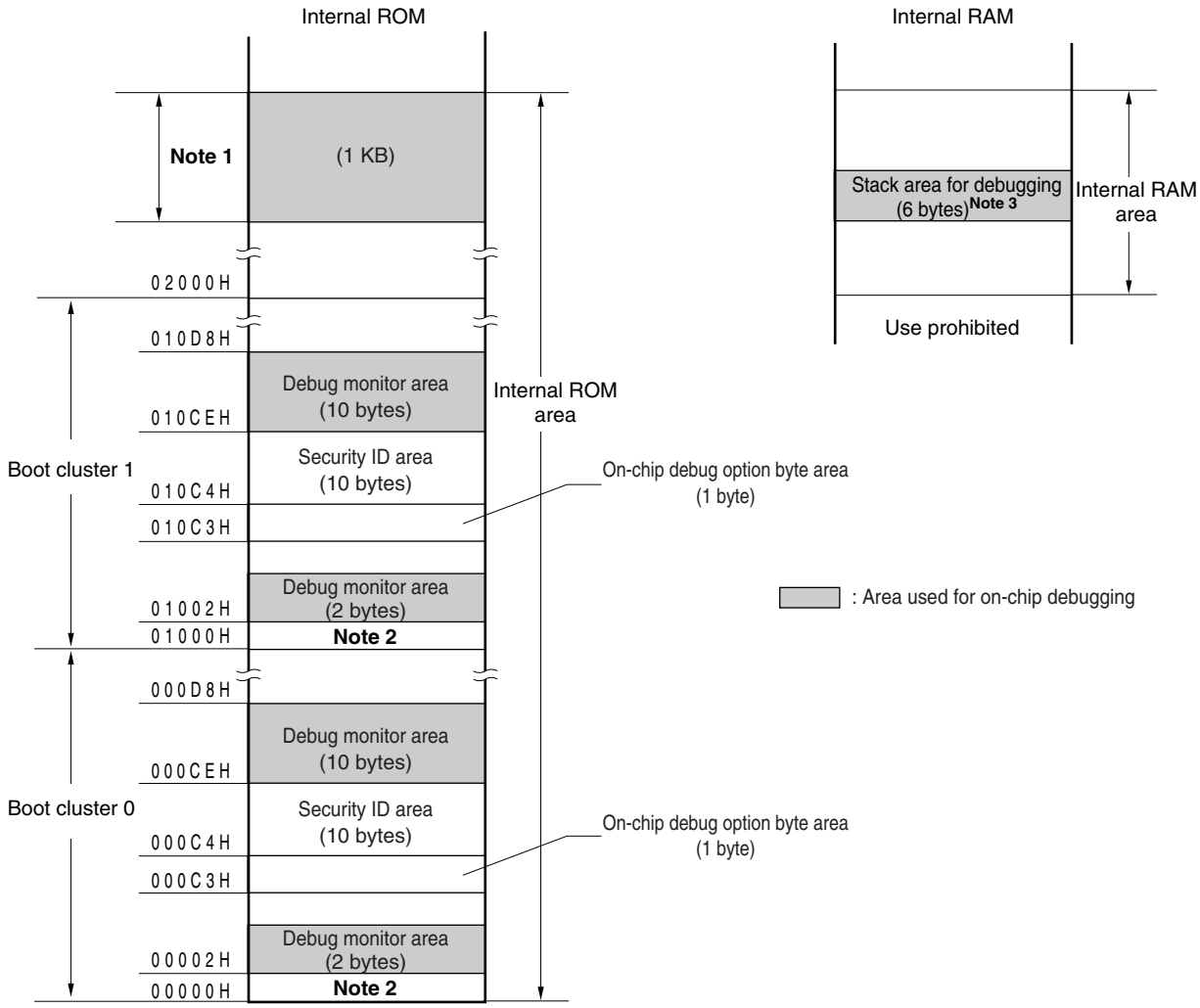
To perform communication between the 78K0R/Kx3-C and QB-MINI2, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler RA78K0R or compiler CC78K0R is used, the items can be set by using linker options.

(1) Securing 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
μPD78F1846A, 78F1848A	96 KB	17C00H to 17FFFH
μPD78F1847A, 78F1849A	128 KB	1FC00H to 1FFFFH

2. In debugging, reset vector is rewritten to address allocated to a monitor program.

3. 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

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 rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol (SFR symbol) FFF00H to FFFFFH Special-function register symbol (16-bit manipulatable SFR symbol. Even addresses only[Note]) FFF00H to FFFFFH
saddr saddrp	FFE20H to FFF1FH Immediate data or labels FFE20H to FF1FH Immediate data or labels (even addresses only[Note])
addr20 addr16 addr5	00000H to FFFFFH Immediate data or labels 0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions[Note]) 0080H to 00BFH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

Remark The special function registers can be described to operand sfr as symbols. See **Table 3-5 SFR List** for the symbols of the special function registers.

The extended special function registers can be described to operand !addr16 as symbols. See **Table 3-6 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

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_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
∀	Exclusive logical sum (exclusive OR)
¯	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

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.

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			
		!addr16, #byte	4	1	–	(addr16) ← byte			
		A, r Note 3	1	1	–	A ← r			
		r, A Note 3	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	×	×	×
		A, PSW	2	1	–	A ← PSW			
		PSW, A	2	3	–	PSW ← A	×	×	×
		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_{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.

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 ← (HL + byte)	
		[HL + byte], A	2	1	–	(HL + byte) ← A	
		A, [HL + B]	2	1	4	A ← (HL + B)	
		[HL + B], A	2	1	–	(HL + B) ← A	
		A, [HL + C]	2	1	4	A ← (HL + C)	
		[HL + C], A	2	1	–	(HL + C) ← A	
		word[B], #byte	4	1	–	(B + word) ← byte	
		A, word[B]	3	1	4	A ← (B + word)	
		word[B], A	3	1	–	(B + word) ← A	
		word[C], #byte	4	1	–	(C + word) ← byte	
		A, word[C]	3	1	4	A ← (C + word)	
		word[C], A	3	1	–	(C + word) ← A	
		word[BC], #byte	4	1	–	(BC + word) ← byte	
		A, word[BC]	3	1	4	A ← (BC + word)	
		word[BC], A	3	1	–	(BC + word) ← A	
		[SP + byte], #byte	3	1	–	(SP + byte) ← byte	
		A, [SP + byte]	2	1	–	A ← (SP + byte)	
		[SP + byte], A	2	1	–	(SP + byte) ← A	
		B, saddr	2	1	–	B ← (saddr)	
		B, !addr16	3	1	4	B ← (addr16)	
		C, saddr	2	1	–	C ← (saddr)	
		C, !addr16	3	1	4	C ← (addr16)	
		X, saddr	2	1	–	X ← (saddr)	
		X, !addr16	3	1	4	X ← (addr16)	
		ES:!addr16, #byte	5	2	–	(ES, addr16) ← byte	
		A, ES:!addr16	4	2	5	A ← (ES, addr16)	
		ES:!addr16, A	4	2	–	(ES, addr16) ← A	
		A, ES:[DE]	2	2	5	A ← (ES, DE)	
		ES:[DE], A	2	2	–	(ES, DE) ← A	
		ES:[DE + byte],#byte	4	2	–	((ES, DE) + byte) ← byte	
		A, ES:[DE + byte]	3	2	5	A ← ((ES, DE) + byte)	
		ES:[DE + byte], A	3	2	–	((ES, DE) + byte) ← 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.

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 ← (ES, HL)	
		ES:[HL], A	2	2	–	(ES, HL) ← A	
		ES:[HL + byte],#byte	4	2	–	((ES, HL) + byte) ← byte	
		A, ES:[HL + byte]	3	2	5	A ← ((ES, HL) + byte)	
		ES:[HL + byte], A	3	2	–	((ES, HL) + byte) ← A	
		A, ES:[HL + B]	3	2	5	A ← ((ES, HL) + B)	
		ES:[HL + B], A	3	2	–	((ES, HL) + B) ← A	
		A, ES:[HL + C]	3	2	5	A ← ((ES, HL) + C)	
		ES:[HL + C], A	3	2	–	((ES, HL) + C) ← A	
		ES:word[B], #byte	5	2	–	((ES, B) + word) ← byte	
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)	
		ES:word[B], A	4	2	–	((ES, B) + word) ← A	
		ES:word[C], #byte	5	2	–	((ES, C) + word) ← byte	
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)	
		ES:word[C], A	4	2	–	((ES, C) + word) ← A	
		ES:word[BC], #byte	5	2	–	((ES, BC) + word) ← byte	
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)	
		ES:word[BC], A	4	2	–	((ES, BC) + word) ← A	
		B, ES:!addr16	4	2	5	B ← (ES, addr16)	
		C, ES:!addr16	4	2	5	C ← (ES, addr16)	
		X, ES:!addr16	4	2	5	X ← (ES, addr16)	
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	–	A ←→ r	
		A, saddr	3	2	–	A ←→ (saddr)	
		A, sfr	3	2	–	A ←→ sfr	
		A, !addr16	4	2	–	A ←→ (addr16)	
		A, [DE]	2	2	–	A ←→ (DE)	
		A, [DE + byte]	3	2	–	A ←→ (DE + byte)	
		A, [HL]	2	2	–	A ←→ (HL)	
		A, [HL + byte]	3	2	–	A ←→ (HL + byte)	
		A, [HL + B]	2	2	–	A ←→ (HL + B)	
		A, [HL + C]	2	2	–	A ←→ (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_{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.

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_{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.

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.

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 Note 3	1	1	–	AX ←→ rp			
	ONEW	AX	1	1	–	AX ← 0001H			
		BC	1	1	–	BC ← 0001H			
	CLRW	AX	1	1	–	AX ← 0000H			
		BC	1	1	–	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) + byte	×	×	×
		A, r Note 4	2	1	–	A, CY ← A + r	×	×	×
		r, A	2	1	–	r, CY ← r + A	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A,CY ← A + ((ES, HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A,CY ← A + ((ES, HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A,CY ← A + ((ES, 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 rp = AX

4. 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.

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	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A + r + CY	×	×	×
		r, A	2	1	–	r, CY ← r + A + CY	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr) + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C) + CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16) + CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A + (ES, HL) + CY	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A + ((ES, HL) + byte) + CY	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A + ((ES, HL) + B) + CY	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A + ((ES, HL) + C) + CY	×	×	×
	SUB	A, #byte	2	1	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r	×	×	×
		r, A	2	1	–	r, CY ← r – A	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A – (HL + C)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A – ((ES: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_{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.

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 ← A – byte – CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r – CY	×	×	×
		r, A	2	1	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A – (HL + C) – CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES:addr16) – CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A – (ES:HL) – CY	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A – ((ES:HL) + byte) – CY	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A – ((ES:HL) + B) – CY	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A – ((ES:HL) + C) – CY	×	×	×
	AND	A, #byte	2	1	–	A ← A ∧ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	–	A ← A ∧ r	×		
		r, A	2	1	–	r ← r ∧ A	×		
		A, saddr	2	1	–	A ← A ∧ (saddr)	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	1	4	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	1	4	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	1	4	A ← A ∧ (HL + C)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	A ← A ∧ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	A ← A ∧ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	A ← A ∧ ((ES: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_{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.

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 ← A ∨ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∨ byte	×		
		A, r Note 3	2	1	–	A ← A ∨ r	×		
		r, A	2	1	–	r ← r ∨ A	×		
		A, saddr	2	1	–	A ← A ∨ (saddr)	×		
		A, !addr16	3	1	4	A ← A ∨ (addr16)	×		
		A, [HL]	1	1	4	A ← A ∨ (HL)	×		
		A, [HL + byte]	2	1	4	A ← A ∨ (HL + byte)	×		
		A, [HL + B]	2	1	4	A ← A ∨ (HL + B)	×		
		A, [HL + C]	2	1	4	A ← A ∨ (HL + C)	×		
		A, ES:!addr16	4	2	5	A ← A ∨ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	A ← A ∨ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	A ← A ∨ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	A ← A ∨ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	A ← A ∨ ((ES:HL) + C)	×		
	XOR	A, #byte	2	1	–	A ← A ∨̲ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∨̲ byte	×		
		A, r Note 3	2	1	–	A ← A ∨̲ r	×		
		r, A	2	1	–	r ← r ∨̲ A	×		
		A, saddr	2	1	–	A ← A ∨̲ (saddr)	×		
		A, !addr16	3	1	4	A ← A ∨̲ (addr16)	×		
		A, [HL]	1	1	4	A ← A ∨̲ (HL)	×		
		A, [HL + byte]	2	1	4	A ← A ∨̲ (HL + byte)	×		
		A, [HL + B]	2	1	4	A ← A ∨̲ (HL + B)	×		
		A, [HL + C]	2	1	4	A ← A ∨̲ (HL + C)	×		
		A, ES:!addr16	4	2	5	A ← A ∨̲ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	A ← A ∨̲ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	A ← A ∨̲ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	A ← A ∨̲ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	A ← A ∨̲ ((ES: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_{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.

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	×	×	×
		saddr, #byte	3	1	–	(saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A – r	×	×	×
		r, A	2	1	–	r – A	×	×	×
		A, saddr	2	1	–	A – (saddr)	×	×	×
		A, !addr16	3	1	4	A – (addr16)	×	×	×
		A, [HL]	1	1	4	A – (HL)	×	×	×
		A, [HL + byte]	2	1	4	A – (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A – (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A – (HL + C)	×	×	×
		!addr16, #byte	4	1	4	(addr16) – byte	×	×	×
		A, ES:!addr16	4	2	5	A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A – ((ES:HL) + C)	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	×	×	×
	CMP0	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	4	(addr16) – 00H	×	×	×
		ES:!addr16	4	2	5	(ES:addr16) – 00H	×	×	×
	CMPS	X, [HL + byte]	3	1	4	X – (HL + byte)	×	×	×
		X, ES:[HL + byte]	4	2	5	X – ((ES:HL) + 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_{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.

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	×	×	×
		AX, AX	1	1	–	AX, CY ← AX + AX	×	×	×
		AX, BC	1	1	–	AX, CY ← AX + BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX + DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX + HL	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX + (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX + (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX + (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX + (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX + ((ES:HL) + byte)	×	×	×
	SUBW	AX, #word	3	1	–	AX, CY ← AX – word	×	×	×
		AX, BC	1	1	–	AX, CY ← AX – BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX – DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX – HL	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX – (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX – (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX – (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX – ((ES:HL) + byte)	×	×	×
	CMPW	AX, #word	3	1	–	AX – word	×	×	×
		AX, BC	1	1	–	AX – BC	×	×	×
		AX, DE	1	1	–	AX – DE	×	×	×
		AX, HL	1	1	–	AX – HL	×	×	×
		AX, saddrp	2	1	–	AX – (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX – (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX – (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX – ((ES:HL) + byte)	×	×	×
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_{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.

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 ← r + 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) + 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) + 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) + 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) + 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) + 1	×	×	
	DEC	r	1	1	–	r ← r – 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) – 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) – 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) – 1	×	×	
	INCW	rp	1	1	–	rp ← rp + 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) + 1			
		!addr16	3	2	–	(addr16) ← (addr16) + 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) + 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) + 1			
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) + 1			
	DECW	rp	1	1	–	rp ← rp – 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) – 1			
		!addr16	3	2	–	(addr16) ← (addr16) – 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1			
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) – 1			
Shift	SHR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) × cnt			×
	SHRW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) × cnt			×
	SHL	A, cnt	2	1	–	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) × cnt			×
		B, cnt	2	1	–	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) × cnt			×
		C, cnt	2	1	–	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) × cnt			×
	SHLW	AX, cnt	2	1	–	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) × cnt			×
		BC, cnt	2	1	–	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) × cnt			×
	SAR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) × cnt			×
	SARW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← AX_{15}) × cnt			×

Notes 1. 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.

3. cnt indicates the bit shift count.

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, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1			×
	ROL	A, 1	2	1	–	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1			×
	RORC	A, 1	2	1	–	(CY $\leftarrow A_0$, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1			×
	ROLC	A, 1	2	1	–	(CY $\leftarrow A_7$, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1			×
	ROLWC	AX,1	2	1	–	(CY $\leftarrow AX_{15}$, $AX_0 \leftarrow$ CY, $AX_{m+1} \leftarrow AX_m$) × 1			×
		BC,1	2	1	–	(CY $\leftarrow BC_{15}$, $BC_0 \leftarrow$ CY, $BC_{m+1} \leftarrow BC_m$) × 1			×
Bit manipulate	MOV1	CY, saddr.bit	3	1	–	CY ← (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← sfr.bit			×
		CY, A.bit	2	1	–	CY ← A.bit			×
		CY, PSW.bit	3	1	–	CY ← PSW.bit			×
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		saddr.bit, CY	3	2	–	(saddr).bit ← CY			
		sfr.bit, CY	3	2	–	sfr.bit ← CY			
		A.bit, CY	2	1	–	A.bit ← CY			
		PSW.bit, CY	3	4	–	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	2	–	(HL).bit ← CY			
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit			×
		ES:[HL].bit, CY	3	3	–	(ES, HL).bit ← CY			
	AND1	CY, saddr.bit	3	1	–	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∧ PSW.bit			×
		CY,[HL].bit	2	1	4	CY ← CY ∧ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit			×
	OR1	CY, saddr.bit	3	1	–	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∨ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∨ PSW.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Notes 1. 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.

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 ← CY ⩒ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ⩒ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ⩒ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ⩒ PSW.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ⩒ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ⩒ (ES, HL).bit			×
	SET1	saddr.bit	3	2	–	(saddr).bit ← 1			
		sfr.bit	3	2	–	sfr.bit ← 1			
		A.bit	2	1	–	A.bit ← 1			
		!addr16.bit	4	2	–	(addr16).bit ← 1			
		PSW.bit	3	4	–	PSW.bit ← 1	×	×	×
		[HL].bit	2	2	–	(HL).bit ← 1			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 1			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 1			
	CLR1	saddr.bit	3	2	–	(saddr.bit) ← 0			
		sfr.bit	3	2	–	sfr.bit ← 0			
		A.bit	2	1	–	A.bit ← 0			
		!addr16.bit	4	2	–	(addr16).bit ← 0			
		PSW.bit	3	4	–	PSW.bit ← 0	×	×	×
		[HL].bit	2	2	–	(HL).bit ← 0			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 0			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 0			
	SET1	CY	2	1	–	CY ← 1			1
	CLR1	CY	2	1	–	CY ← 0			0
	NOT1	CY	2	1	–	CY ← $\overline{CY}$			×

Notes 1. 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.

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.

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) ← rp_H, (SP – 2)← rp_L, SP ← SP – 2			
	POP	PSW	2	3	–	PSW ← (SP + 1), SP ← SP + 2	R	R	R
		rp	1	1	–	rp_L ← (SP), rp_H ← (SP + 1), SP ← SP + 2			
	MOVW	SP, #word	4	1	–	SP ← word			
		SP, AX	2	1	–	SP ← AX			
		AX, SP	2	1	–	AX ← SP			
		HL, SP	3	1	–	HL ← SP			
		BC, SP	3	1	–	BC ← SP			
		DE, SP	3	1	–	DE ← SP			
	ADDW	SP, #byte	2	1	–	SP ← SP + byte			
	SUBW	SP, #byte	2	1	–	SP ← SP – byte			
Unconditional branch	BR	AX	2	3	–	PC ← CS, AX			
		$addr20	2	3	–	PC ← PC + 2 + jdisp8			
		$!addr20	3	3	–	PC ← PC + 3 + jdisp16			
		!addr16	3	3	–	PC ← 0000, addr16			
		!!addr20	4	3	–	PC ← addr20			
Conditional branch	BC	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4[Note 3]	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=0			
	BNH	$addr20	3	2/4[Note 3]	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=1			
	BT	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5[Note 3]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

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.

Table 28-5. Operation List (17/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5[Note 3]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6[Note 3]	–	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit			
Conditional skip	SKC	–	2	1	–	Next instruction skip if CY = 1			
	SKNC	–	2	1	–	Next instruction skip if CY = 0			
	SKZ	–	2	1	–	Next instruction skip if Z = 1			
	SKNZ	–	2	1	–	Next instruction skip if Z = 0			
	SKH	–	2	1	–	Next instruction skip if (Z ∨ CY) = 0			
	SKNH	–	2	1	–	Next instruction skip if (Z ∨ CY) = 1			
CPU control	SEL	RBn	2	1	–	RBS[1:0] ← n			
	NOP	–	1	1	–	No Operation			
	EI	–	3	4	–	IE ← 1(Enable Interrupt)			
	DI	–	3	4	–	IE ← 0(Disable Interrupt)			
	HALT	–	2	3	–	Set HALT Mode			
	STOP	–	2	3	–	Set STOP Mode			

Notes 1. 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.

3. n indicates the number of register banks (n = 0 to 3)

CHAPTER 29 ELECTRICAL SPECIFICATIONS

Caution The 78K0R/Kx3-C has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.5 to +6.5	V
	EV_{DD0}, EV_{DD1}	EV_{DD0} = EV_{DD1}	−0.5 to +6.5	V
	V_{SS}		−0.5 to +0.3	V
	EV_{SS0}, EV_{SS1}	EV_{SS0} = EV_{SS1}	−0.5 to +0.3	V
	AV_{REF}		−0.5 to V_{DD} +0.3 Note 1	V
	AV_{SS}		−0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC	−0.3 to +3.6 and −0.3 to V_{DD} +0.3 Note 2	V
Input voltage	V_{I1}	P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111, P120 to P124, P131, P140 to P145, EXCLK, $\overline{\text{RESET}}$, FLMD0	−0.3 to EV_{DD0}, EV_{DD1} +0.3 and −0.3 to V_{DD} +0.3 Note 1	V
	V_{I2}	P60 to P63 (N-ch open-drain)	−0.3 to +6.5	V
	V_{I3}	P20 to P27, P150 to P157	−0.3 to AV_{REF} +0.3 and −0.3 to V_{DD} +0.3 Note 1	V
Output voltage	V_{O1}	P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111, P120, P130, P131, P140 to P145	−0.3 to EV_{DD0}, EV_{DD1} +0.3 Note 1	V
	V_{O2}	P20 to P27, P150 to P157	−0.3 to AV_{REF} +0.3	V

Notes 1. Must be 6.5 V or lower.

2. Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. The presence or absence of the following port pins depends on the product.

P90: 78K0R/KF3-C only

P00, P01, P56, P57, P80 to P87, P131, P141, P145, P154 to P157: 78K0R/KG3-C only

3. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbols	Conditions		Ratings	Unit
Analog input voltage	V_{AN}	ANI0 to ANI15		−0.3 to AV_{REF} +0.3[Note] and −0.3 to V_{DD} +0.3[Note]	V
Output current, high	I_{OH1}	Per pin	P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111, P120, P130, P131, P140 to P145	−10	mA
		Total of all pins −80 mA	P00 to P04, P40 to P47, P120, P130, P131, P140 to P145	−25	mA
			P05, P06, P10 to P17, P30, P31, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111	−55	mA
	I_{OH2}	Per pin	P20 to P27, P150 to P157	−0.5	mA
		Total of all pins		−2	mA
Output current, low	I_{OL1}	Per pin	P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111, P120, P130, P131, P140 to P145	30	mA
		Total of all pins 200 mA	P00 to P04, P40 to P47, P120, P130, P131, P140 to P145	60	mA
			P05, P06, P10 to P17, P30, P31, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111	140	mA
	I_{OL2}	Per pin	P20 to P27, P150 to P157	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A	In normal operation mode		−40 to +85	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			−65 to +150	°C

Note Must be 6.5 V or lower.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. The presence or absence of the following port pins depends on the product.

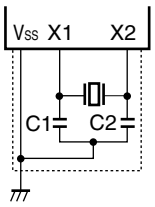
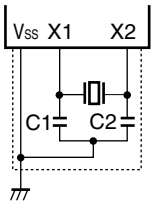
P90: 78K0R/KF3-C only

P00, P01, P56, P57, P80 to P87, P131, P141, P145, P154 to P157: 78K0R/KG3-C only

3. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

X1 Oscillator Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency (fx)Note	2.7 V ≤ V_{DD} ≤ 5.5 V	2.0		20.0	MHz
Crystal resonator		X1 clock oscillation frequency (fx)Note	2.7 V ≤ V_{DD} ≤ 5.5 V	2.0		20.0	MHz

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.

Remarks 1. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

2. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Internal Oscillator Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
Internal high-speed oscillation clock frequency Note	f_{IH8}	2.7 V ≤ V_{DD} ≤ 5.5 V		7.856	8.0	8.144	MHz
	f_{IH20}	2.7 V ≤ V_{DD} ≤ 5.5 V		19.52	20.0	20.48	MHz
Internal low-speed oscillation clock frequency	f_{IL}	2.7 V ≤ V_{DD} ≤ 5.5 V	Normal power consumption mode (RMC = 00H)	27	30	33	kHz
			Low power consumption mode (RMC = 5AH)	25.5	30	34.5	kHz

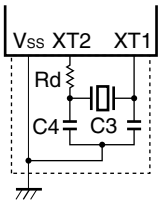
Note This only indicates the oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Remarks 1. RMC: Regulator mode control register

2. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

XT1 Oscillator Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Items	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency (f_{XT})Note		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_{SS}.**
- **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.

Remarks 1. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

2. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

Recommended oscillator circuit constants

(1) X1 oscillation: Ceramic resonator (AMPH = 0, RMC = 00H, T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants			Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
Murata Manufacturing Co., Ltd. Note	CSTCC2M00G56-R0	SMD	2.0	Internal (47)	Internal (47)	0	2.7	5.5
	CSTCR4M00G55-R0	SMD	4.0	Internal (39)	Internal (39)	0		
	CSTLS4M00G56-B0	Lead		Internal (47)	Internal (47)	0		
	CSTCR4M19G55-R0	SMD	4.194	Internal (39)	Internal (39)	0		
	CSTLS4M19G56-B0	Lead		Internal (47)	Internal (47)	0		
	CSTCR4M91G55-R0	SMD	4.915	Internal (39)	Internal (39)	0		
	CSTLS4M91G56-B0	Lead		Internal (47)	Internal (47)	0		
	CSTCR5M00G55-R0	SMD	5.0	Internal (39)	Internal (39)	0		
	CSTLS5M00G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCR6M00G53-R0	SMD	6.0	Internal (15)	Internal (15)	0		
	CSTLS6M00G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCE8M00G55-R0	SMD	8.0	Internal (33)	Internal (33)	0		
	CSTLS8M00G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCE8M38G55-R0	SMD	8.388	Internal (33)	Internal (33)	0		
	CSTLS8M38G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCE10M0G52-R0	SMD	10.0	Internal (10)	Internal (10)	0		
	CSTLS10M0G53-B0	Lead		Internal (15)	Internal (15)	0		

(2) X1 oscillation: Crystal resonator (AMPH = 0, RMC = 00H, T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants			Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
KYOCERA KINSEKI Corporation	HC49SFWB	Lead	4.0	10	10	0	1.8	5.5
	HC49SFWB	Lead	4.9152	10	10	0		
	HC49SFWB	Lead	5.0	10	10	0		
	HC49SFWB	Lead	6.0	10	10	0		
	HC49SFWB	Lead	8.0	10	10	0		
	HC49SFWB	Lead	8.38	10	10	0		
	HC49SFWB	Lead	10.0	10	10	0		

(3) X1 oscillation: Ceramic resonator (AMPH = 1, RMC = 00H, T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants			Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
Murata Manufacturing Co., Ltd.. Note	CSTCE12M0G55-R0	SMD	12.0	Internal (33)	Internal (33)	0	2.7	5.5
	CSTCE16M0V53-R0	SMD	16.0	Internal (15)	Internal (15)	0		
	CSTLS16M0X51-B0	Lead		Internal (5)	Internal (5)	0		
	CSTCE20M0V53-R0	SMD	20.0	Internal (15)	Internal (15)	0		
	CSTLS20M0X51-B0	Lead		Internal (5)	Internal (5)	0		

Note Contact Murata Manufacturing Co., Ltd. (http://www.murata.com) when using this resonator.

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. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KF3-C, 78K0R/KG3-C so that the internal operation conditions are within the specifications of the DC and AC characteristics.

(4) X1 oscillation: Crystal resonator (AMPH = 1, RMC = 00H, T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants			Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
KYOCERA KINSEKI Corporation	HC49SFWB	Lead	12.0	10	10	0	1.8	5.5
	HC49SFWB	Lead	16.0	10	10	0		
	HC49SFWB	Lead	20.0	10	10	0		

(5) X1 oscillation: Ceramic resonator (AMPH = 1, RMC = 5AH, T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants			Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
Murata Manufacturing Co., Ltd. Note 1	CSTCC2M00G56-R0	SMD	2.0	Internal (47)	Internal (47)	0	2.7	5.5
	CSTCR4M00G55-R0	SMD	4.0	Internal (39)	Internal (39)	0		
	CSTLS4M00G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCR4M19G55-R0	SMD	4.194	Internal (39)	Internal (39)	0		
	CSTLS4M19G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCR4M91G55-R0	SMD	4.915	Internal (39)	Internal (39)	0		
	CSTLS4M91G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCR5M00G53-R0	SMD	5.0	Internal (15)	Internal (15)	0		
	CSTLS5M00G53-B0	Lead		Internal (15)	Internal (15)	0		

(6) XT1 oscillation: Crystal resonator (T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Load Capacitance CL (pF)	XT1 oscillator oscillation mode Note 1	Recommended Circuit Constants			Oscillation Voltage Range	
						C3 (pF)	C4 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
CITIZEN FINETECH MIYOTA CO., LTD. Note 2	CMR200T	SMD	32.768	7	Normal oscillation	9	9	0	1.8	5.5
				7	Low power consumption oscillation	9	9	0		
				7	Ultra-low power consumption oscillation	9	9	0		
SEIKO INSTRUMENTS INC Note 3	SSP-T7-FL	SMD	32.768	6.0	Normal oscillation	9	8	0	1.8	5.5
				4.4	Low power consumption oscillation	6	5	0		
				3.7	Ultra-low power consumption oscillation	4	4	0		
	VT-200-FL	Lead		6.0	Normal oscillation	9	8	0		
				4.4	Low power consumption oscillation	6	5	0		
				3.7	Ultra-low power consumption oscillation	4	4	0		

Notes 1. Contact Murata Manufacturing Co., Ltd. (http://www.murata.com) when using this resonator.
2. Set the XT1 oscillation mode by using bits AMPHS1 and AMPHS0 of the clock operation mode control register (CMC).
3. Contact CITIZEN FINETECH MIYOTA CO., LTD. (http://cfm.citizen.co.jp) when using this resonator.
4. Contact SEIKO INSTRUMENTS INC. (http://www.sii-crystal.com) when using this resonator.

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. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/KF3-C, 78K0R/KG3-C 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°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111, P120, P130, P131, P140 to P145	4.0 V ≤ V_{DD} ≤ 5.5 V			–3.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–1.0	mA
		Total of P00 to P04, P40 to P47, P120, P130, P131, P140 to P145 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			–20.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–10.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			–30.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–19.0	mA
		Total of all pins (When duty = 60%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			–50.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–29.0	mA
	I_{OH2}	Per pin for P20 to P27, P150 to P157	AV_{REF} = 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} × 0.7)/(n × 0.01)

<Example> Where n = 50% and I_{OH} = –20.0 mA

Total output current of pins = (–20.0 × 0.7)/(50 × 0.01) = –28.0 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Caution P02 to P04, P10, P12, P40, P142 to P144 do not output high level in N-ch open-drain mode.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. The presence or absence of the following port pins depends on the product.

P90: 78K0R/KF3-C only

P00, P01, P56, P57, P80 to P87, P131, P141, P145, P154 to P157: 78K0R/KG3-C only

3. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

DC Characteristics (2/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111, P120, P130, P131, P140 to P145	4.0 V ≤ V_{DD} ≤ 5.5 V			8.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			1.0	mA
		Per pin for P60 to P63	4.0 V ≤ V_{DD} ≤ 5.5 V			15.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			3.0	mA
		Total of P00 to P04, P40 to P47, P120, P130, P131, P140 to P145 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			20.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			15.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			45.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			35.0	mA
		Total of all pins (When duty = 60%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			65.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			50.0	mA
	I_{OL2}	Per pin for P20 to P27, P150 to P157	AV_{REF} = 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}, and 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 = (I_{OL} × 0.7)/(n × 0.01)

<Example> Where n = 50% and I_{OL} = 20.0 mA

Total output current of pins = (20.0 × 0.7)/(50 × 0.01) = 28.0 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. The presence or absence of the following port pins depends on the product.

P90: 78K0R/KF3-C only

P00, P01, P56, P57, P80 to P87, P131, P141, P145, P154 to P157: 78K0R/KG3-C only

3. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

DC Characteristics (3/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P01, P02, P12, P13, P15, P41, P45, P52, P56, P57, P80 to P87, P90, P91, P111, P123, P124, 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, P53 to P55, P64 to P67, P70 to P77, P110, P120 to P122, P131, P140 to P143, P145, EXCLK, $\overline{\text{RESET}}$	Normal input buffer	0.8V_{DD}		V_{DD}	V
	V_{IH3}	P03, P04, P10, P11, P142, P143	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	2.2		V_{DD}	V
			TTL input buffer 2.7 V ≤ V_{DD} < 4.0 V	2.0		V_{DD}	V
	V_{IH4}	P20 to P27, P150 to P157	AV_{REF} = V_{DD}	0.7AV_{REF}		AV_{REF}	V
	V_{IH5}	P60 to P63		0.7V_{DD}		6.0	V
	V_{IH6}	P62	CEC input buffer V_{DD} = 3.3 V±10%	2.0		3.63	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 1. The maximum value of V_{IH} of pins P02 to P04, P10, P12, P40, and P142 to P144 is V_{DD}, even in the N-ch open-drain mode.

2. For P122/EXCLK, the value of V_{IH} and V_{IL} differs according to the input port mode or external clock mode.
Make sure to satisfy the DC characteristics of EXCLK in external clock input mode.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. The presence or absence of the following port pins depends on the product.
P90: 78K0R/KF3-C only
P00, P01, P56, P57, P80 to P87, P131, P141, P145, P154 to P157: 78K0R/KG3-C only

3. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

DC Characteristics (4/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, low	V_{IL1}	P01, P02, P12, P13, P15, P41, P45, P52, P56, P57, P80 to P87, P90, P91, P111, P123, P124, 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, P53 to P55, P64 to P67, P70 to P77, P110, P120 to P122, P131, P140 to P143, P145, EXCLK, $\overline{\text{RESET}}$	Normal input buffer	0		$0.2V_{DD}$	V
	V_{IL3}	P03, P04, P10, P11, P142, P143	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 2.7 V ≤ V_{DD} < 4.0 V	0		0.5	V
	V_{IL4}	P20 to P27, P150 to P157	AV_{REF} = V_{DD}	0		$0.3AV_{REF}$	V
	V_{IL5}	P60 to P63		0		$0.3V_{DD}$	V
	V_{IL6}	P62	CEC input buffer V_{DD} = 3.3 V±10%	0		0.8	V
	V_{IL7}	FLMD0 Note		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 The maximum value of V_{IH} of pins P02 to P04, P10, P12, P40, and P142 to P144 is V_{DD}, even in the N-ch open-drain mode.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. The presence or absence of the following port pins depends on the product.

P90: 78K0R/KF3-C only

P00, P01, P56, P57, P80 to P87, P131, P141, P145, P154 to P157: 78K0R/KG3-C only

3. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

DC Characteristics (5/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111, P120, P130, P131, P140 to P145	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = – 3.0 mA	V_{DD} – 0.7			V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –1.0 mA	V_{DD} – 0.5			V
	V_{OH2}	P20 to P27, P150 to P157	AV_{REF} = V_{DD}, I_{OH2} = –0.1 mA	AV_{REF} – 0.5			V
	V_{OH3}	P62	CECIO mode [Note] V_{DD} = 3.3 V±10% I_{OH1} = –12.0 μA	2.5		3.63	V
Output voltage, low	V_{OL1}	P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111, P120, P130, P131, P140 to P145	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.0 mA			0.5	V
	V_{OL2}	P20 to P27, P150 to P157	AV_{REF} = V_{DD}, I_{OL2} = 0.4 mA			0.4	V
	V_{OL3}	P60 to P63	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 15.0 mA			2.0	V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 5.0 mA			0.4	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.4	V
	V_{OL4}	P62	CECIO mode V_{DD} = 3.3 V±10% I_{OL1} = 3.3 mA			0.6	V

Note When a pull-up resistor and a diode are connected.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. The presence or absence of the following port pins depends on the product.

P90: 78K0R/KF3-C only

P00, P01, P56, P57, P80 to P87, P131, P141, P145, P154 to P157: 78K0R/KG3-C only

3. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

DC Characteristics (6/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111, P120, P131, P140 to P145, FLMD0, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P20 to P27, P150 to P157	$V_I = V_{SS}$, $AV_{REF} = V_{DD}$				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, XT1, XT2)	$V_I = V_{DD}$	In input port			1	μA
				In resonator connection			10	μA
	I_{LIH4}	P62	CECIO mode V_I = 3.63 V				1.8	μA
Input leakage current, low	I_{LIL1}	P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111, P120, P131, P140 to P145, FLMD0, $\overline{\text{RESET}}$	$V_I = V_{SS}$				–1	μA
	I_{LIL2}	P20 to P27, P150 to P157	$V_I = V_{SS}$, $AV_{REF} = V_{DD}$				–1	μA
	I_{LIL3}	P121 to P124 (X1, X2, XT1, XT2)	$V_I = V_{SS}$	In input port			–1	μA
				In resonator connection			–10	μA
	I_{LIL4}	P62	CECIO mode $V_I = V_{SS}$				–1.8	μA

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. The presence or absence of the following port pins depends on the product.

P90: 78K0R/KF3-C only

P00, P01, P56, P57, P80 to P87, P131, P141, P145, P154 to P157: 78K0R/KG3-C only

3. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

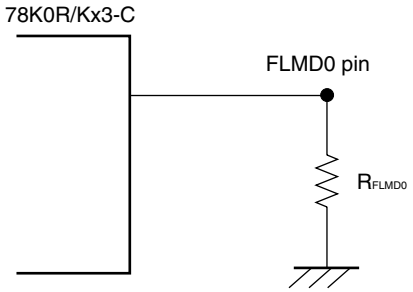
DC Characteristics (7/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
On-chip pll-up resistance	R_U	P00 to P06, P10 to P17, P30, P31, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90, P91, P110, P111, P120, P131, P140 to P145	V_I = V_{SS}, In input port	10	20	100	kΩ
		P62	CECIO mode Note 1	18	26	28.6	kΩ
FLMD0 pin external pull-down resistance Note 2	R_{FLMD0}	When enabling the self-programming mode setting with software		100			kΩ

Notes 1. When a pull-up resistor and a diode are connected.

2. It is recommended to leave the FLMD0 pin open. If the pin is required to be pulled down externally, set R_{FLMD0} to 100 kΩ or more.



Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. The presence or absence of the following port pins depends on the product.

P90: 78K0R/KF3-C only

P00, P01, P56, P57, P80 to P87, P131, P141, P145, P154 to P157: 78K0R/KG3-C only

3. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

DC Characteristics (8/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD1} Note 1	Operating mode	f_{MX} = 20 MHz Note 2, V_{DD} = 5.0 V	Square wave input		5.9	8.3	mA
				Resonator connection		6.2	8.6	mA
			f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V	Square wave input		5.9	8.3	mA
				Resonator connection		6.2	8.6	mA
			f_{MX} = 10 MHz Notes 2, 3, V_{DD} = 5.0 V	Square wave input		3.3	4.8	mA
				Resonator connection		3.4	4.9	mA
			f_{MX} = 10 MHz Notes 2, 3, V_{DD} = 3.0 V	Square wave input		3.3	4.8	mA
				Resonator connection		3.4	4.9	mA
			f_{MX} = 5 MHz Notes 2, 3, V_{DD} = 3.0 V	Square wave input		1.8	2.7	mA
				Resonator connection		1.9	2.8	mA
			f_{IH} = 20 MHz Note 4	V_{DD} = 5.0 V		6.1	8.6	mA
				V_{DD} = 3.0 V		6.1	8.6	mA
			f_{IH} = 8 MHz Note 4	V_{DD} = 5.0 V		2.6	3.8	mA
				V_{DD} = 3.0 V		2.6	3.8	mA

Notes 1. Total current flowing into V_{DD}, EV_{DD0}, EV_{DD1}, and AV_{REF}, 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, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors.

2. When internal high-speed oscillator, 20 MHz internal high-speed oscillator, and subsystem clock are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0, FLPC and FSEL (bits 1 and 0 of operation speed mode control register (OSMC)) = 0 and 0.

4. When high-speed system clock and subsystem clock are stopped.

Remarks 1. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

2. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

f_{IH20}: 20 MHz internal high-speed oscillation clock frequency

f_{IH}: Internal high-speed oscillation clock frequency

3. Temperature condition of the TYP. value is T_A = 25°C

DC Characteristics (9/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD1} Note 1	Operating mode	f_{SUB} = 32.768 kHz Note 2, T_A = –40 to +50 °C	V_{DD} = 5.0 V		4.3	9.3	μA
				V_{DD} = 3.0 V		4.3	9.3	μA
			f_{SUB} = 32.768 kHz Note 2, T_A = –40 to +70 °C	V_{DD} = 5.0 V		4.3	12.3	μA
				V_{DD} = 3.0 V		4.3	12.3	μA
			f_{SUB} = 32.768 kHz Note 2, T_A = –40 to +85 °C	V_{DD} = 5.0 V		4.3	15.5	μA
				V_{DD} = 3.0 V		4.3	15.5	μA

Notes 1. Total current flowing into V_{DD}, EV_{DD0}, EV_{DD1}, and AV_{REF}, 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, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors.

2. When internal high-speed oscillation, 20 MHz internal high-speed oscillation, and high-speed system clock are stopped. When watchdog timer is stopped.

Remarks 1. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

3. Temperature condition of the TYP. value is T_A = 25°C

DC Characteristics (10/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD2} Note 1	HALT mode	f_{MX} = 20 MHz Note 2, V_{DD} = 5.0 V	Square wave input		1.2	3.6	mA
				Resonator connection		1.5	3.9	mA
			f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V	Square wave input		1.2	3.6	mA
				Resonator connection		1.5	3.9	mA
			f_{MX} = 10 MHz Notes 2, 3, V_{DD} = 5.0 V	Square wave input		0.7	2.1	mA
				Resonator connection		0.8	2.2	mA
			f_{MX} = 10 MHz Notes 2, 3, V_{DD} = 3.0 V	Square wave input		0.7	2.1	mA
				Resonator connection		0.8	2.2	mA
			f_{MX} = 5 MHz Notes 2, 3, V_{DD} = 3.0 V	Square wave input		0.41	1.8	mA
				Resonator connection		0.46	1.8	mA
			f_{IH20} = 20 MHz Note 4	V_{DD} = 5.0 V		1.4	3.9	mA
				V_{DD} = 3.0 V		1.4	3.9	mA
			f_{IH} = 8 MHz Note 4	V_{DD} = 5.0 V		0.48	1.8	mA
				V_{DD} = 3.0 V		0.48	1.8	mA

Notes 1. Total current flowing into V_{DD}, EV_{DD0}, EV_{DD1}, and AV_{REF}, 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, 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, 20 MHz internal high-speed oscillator, and subsystem clock are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0, FLPC and FSEL (bits 1 and 0 of operation speed mode control register (OSMC)) = 0 and 0.

4. When high-speed system clock and subsystem clock are stopped.

Remarks 1. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

2. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)
f_{IH20}: 20 MHz internal high-speed oscillation clock frequency
f_{IH}: Internal high-speed oscillation clock frequency

3. Temperature condition of the TYP. value is T_A = 25°C

DC Characteristics (11/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD2} Note 1	HALT mode	f_{SUB} = 32.768 kHz Note 2, T_A = –40 to +50 °C	V_{DD} = 5.0 V		1.0	3.7	μA
				V_{DD} = 3.0 V		1.0	3.7	μA
			f_{SUB} = 32.768 kHz Note 2, T_A = –40 to +70 °C	V_{DD} = 5.0 V		1.0	6.1	μA
				V_{DD} = 3.0 V		1.0	6.1	μA
			f_{SUB} = 32.768 kHz Note 2, T_A = –40 to +85 °C	V_{DD} = 5.0 V		1.0	8.9	μA
				V_{DD} = 3.0 V		1.0	8.9	μA
	I_{DD3} Note 3	STOP mode	T_A = –40 to +50 °C			0.37	2.8	μA
			T_A = –40 to +70 °C			0.37	5.2	μA
			T_A = –40 to +85 °C			0.37	7.9	μA

Notes 1. Total current flowing into V_{DD}, EV_{DD0}, EV_{DD1}, and AV_{REF}, 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, 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 oscillation, 20 MHz internal high-speed oscillation, and high-speed system clock are stopped. When watchdog timer is stopped. When RTCLPC = 1 (stops supply of subsystem clock to peripheral functions other than real-time counter).

3. Total current flowing into V_{DD}, EV_{DD0}, EV_{DD1}, and AV_{REF}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The maximum value includes the peripheral operation current and STOP leakage current. However, not including the current flowing into the A/D converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors. When subsystem clock is stopped. When watchdog timer is stopped.

Remarks 1. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

3. RTCLPC: bit 7 of the operation speed mode control register (OSMC)

4. Temperature condition of the TYP. value is T_A = 25°C

DC Characteristics (12/12)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
RTC operating current	I_{RTC} Notes 1, 2	f_{SUB} = 32.768 kHz		V_{DD} = 3.0 V		0.2	1.0	μA
Watchdog timer operating current	I_{WDT} Notes 2, 3	f_{IL} = 30 kHz				0.31	0.35	μA
A/D converter operating current	I_{ADC} Note 4	During conversion at maximum speed	High speed mode 1	AV_{REF} = V_{DD} = 5.0 V		1.72	3.2	mA
			High speed mode 2	AV_{REF} = V_{DD} = 3.0 V		0.72	1.6	mA
			Normal mode	AV_{REF} = V_{DD} = 5.0 V		0.86	1.9	mA
			Low voltage mode	AV_{REF} = V_{DD} = 3.0 V		0.37	0.8	mA
LVI operating current	I_{LVI} Note 5					9	18	μA

Notes 1. Current flowing only to the real-time counter (excluding the operating current of the XT1 oscillator). The TYP. value of the current value of the 78K0R/Kx3-C is 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 the real-time counter operates during f_{CLK} = f_{SUB}/2, the TYP. value of I_{DD2} includes the real-time counter operating current.

2. When internal high-speed oscillator and high-speed system clock are stopped.

3. Current flowing only to the watchdog timer (including the operating current of the 30 kHz internal oscillator). The current value of the 78K0R/Kx3-C is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer operates during f_{CLK} = f_{SUB}/2 or STOP mode.

4. Current flowing only to the A/D converter (AV_{REF} pin). The current value of the 78K0R/Kx3-C 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.

5. Current flowing only to the LVI circuit. The current value of the 78K0R/Kx3-C 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 1. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

2. f_{IL}: Internal low-speed oscillation clock frequency
f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)
f_{CLK}: CPU/peripheral hardware clock frequency

3. Temperature condition of the TYP. value is T_A = 25°C

AC Characteristics

(1) Basic operation (1/6)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	Normal power consumption mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.05		8	μs
			Low power consumption mode		1		8	μs
		Subsystem clock (f_{SUB}) operation			57.2	61	62.5	μs
		In the self programming mode	Normal power consumption mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.05		1	μs
			Low consumption current mode			1		μs
External main system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V	Normal power consumption mode		2.0		20.0	MHz
			Low power consumption mode		2.0		5.0	MHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V	Normal power consumption mode		24			ns
			Low power consumption mode		96			ns
TI00 to TI07, TI10 to TI12 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK}+10			ns
TO00 to TO07, TO10 to TO12 output frequency	f_{TO}	2.7 V ≤ V_{DD} ≤ 5.5 V					10	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	2.7 V ≤ V_{DD} ≤ 5.5 V					10	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}				1			μs
Key interrupt input low-level width	t_{KR}				250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

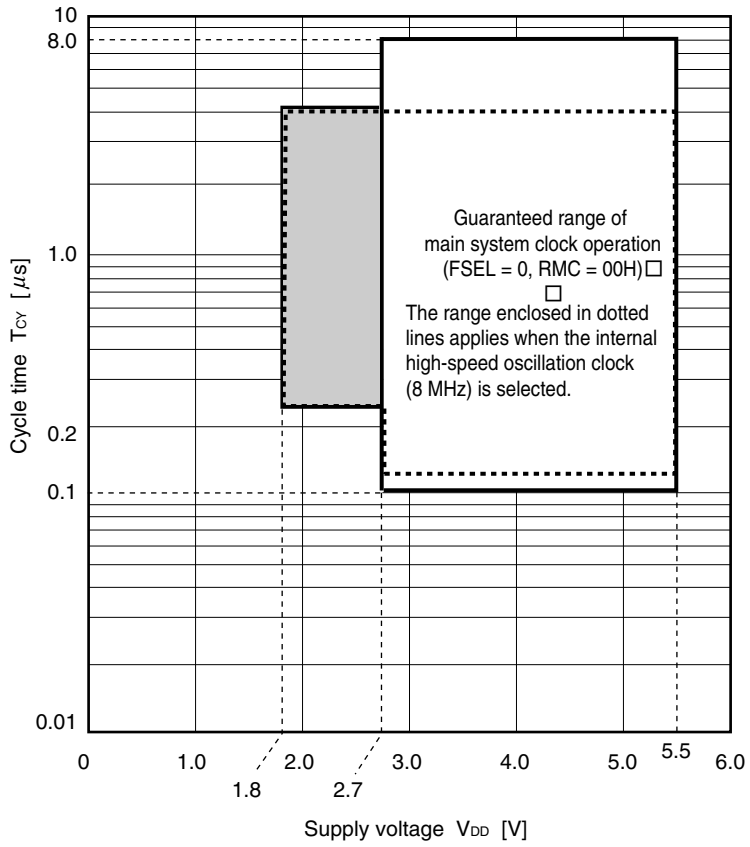
Remarks 1. f_{MCK}: Macro operation clock frequency
(Operation clock to be set by the CKSmn bit of the TMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7))

2. For details on the normal power consumption mode and low power consumption mode according to the regulator output voltage, refer to **CHAPTER 23 REGULATOR**.

3. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

(1) Basic operation (2/6)

Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 00H)



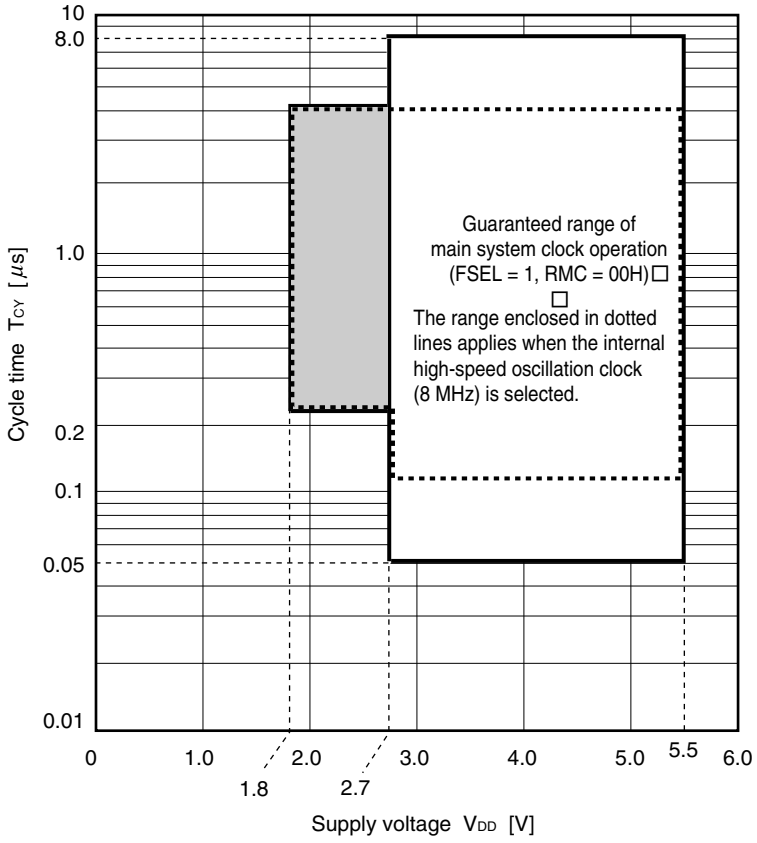
Caution Only the following operations are possible in the shaded section above:

- **Using the CPU (executing instructions)**
  The CPU clock is a fraction of the 8 MHz internal high-speed oscillation clock ($f_{IH}/2^5$ to $f_{IH}/2$ (250 kHz to 4 MHz)).
- **Reading or writing the internal RAM**
- **Using the low-voltage detector (LVI)**
- **Using the interval timer function of the timer array unit (TAU)**
- **Specifying the mode of the standby function (STOP or HALT mode)**
- **Setting up the control registers of the clock generator (except those of the clock and buzzer output controllers)**
  However, the clock can be switched only if the clock cycle time after the clock has been switched will be within the guaranteed operating range.
- **Using the watchdog timer (WDT) (including the internal low-speed oscillator)**

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)



Cautions 1. To set the FSEL bit to 0, set f_{CLK} to 10 MHz or less in advance.

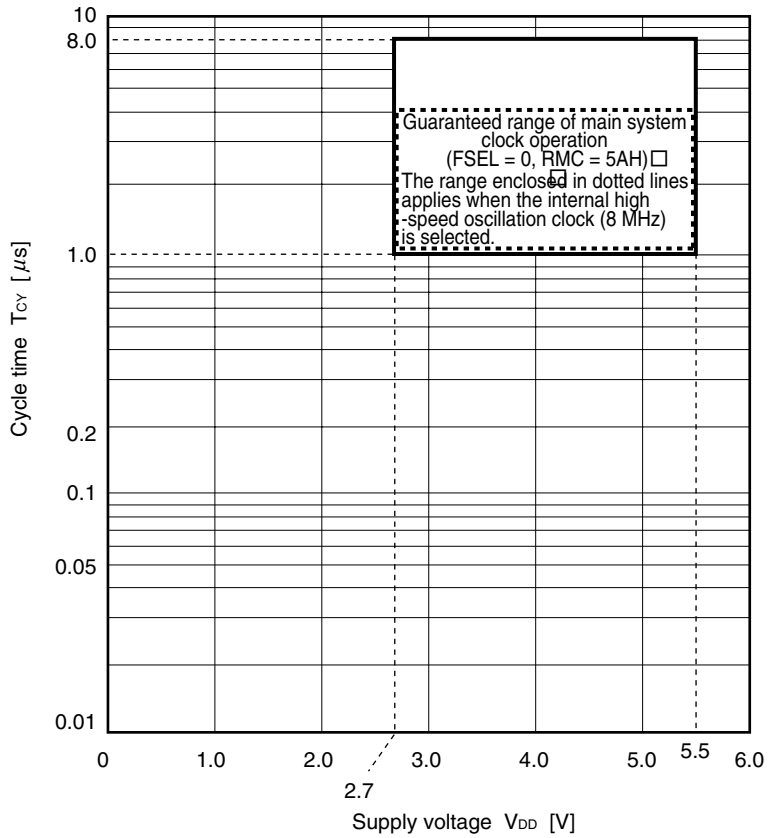
2. Only the following operations are possible in the shaded section above:

- **Using the CPU (executing instructions)**
  The CPU clock is a fraction of the 8 MHz internal high-speed oscillation clock ($f_{IH}/2^5$ to $f_{IH}/2$ (250 kHz to 4 MHz)).
- **Reading or writing the internal RAM**
- **Using the low-voltage detector (LVI)**
- **Using the interval timer function of the timer array unit (TAU)**
- **Specifying the mode of the standby function (STOP or HALT mode)**
- **Setting up the control registers of the clock generator (except those of the clock and buzzer output controllers)**
  However, the clock can be switched only if the clock cycle time after the clock has been switched will be within the guaranteed operating range.
- **Using the watchdog timer (WDT) (including the internal low-speed oscillator)**

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)

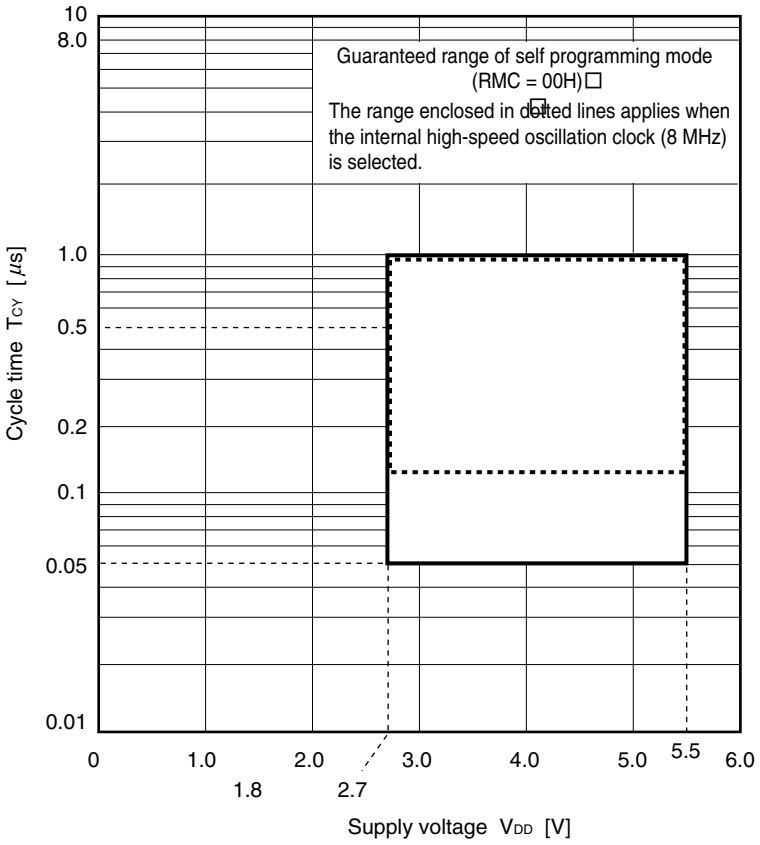


Remarks 1. FSEL: Bit 0 of the operation speed mode control register (OSMC)
RMC: Regulator mode control register

2. The entire voltage range is 1 MHz (MAX.) when RMC is set to 5AH.

(1) Basic operation (5/6)

Minimum instruction execution time during self programming mode (RMC = 00H)



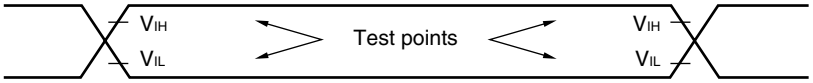
Remarks 1. RMC: Regulator mode control register

2. The self programming function cannot be used when RMC is set to 5AH or the CPU operates with the subsystem clock.

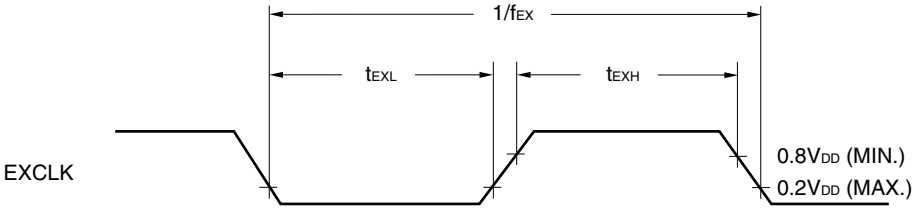
3. The entire voltage range is 1 MHz when RMC is set to 5AH.

(1) Basic operation (6/6)

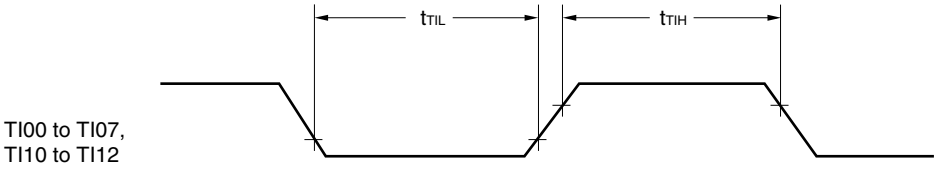
AC Timing Test Points (Excluding external bus interface)



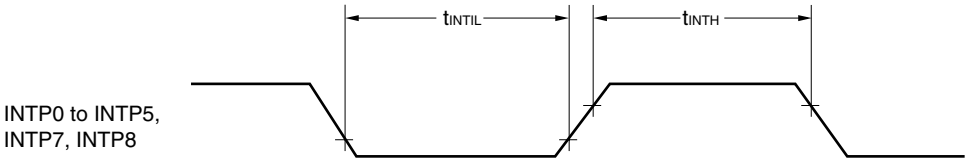
External Main System Clock Timing



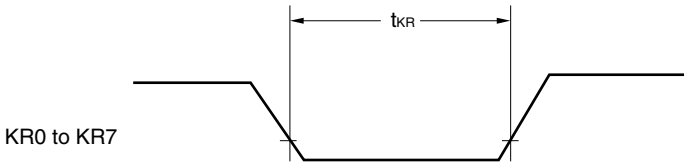
TI Timing



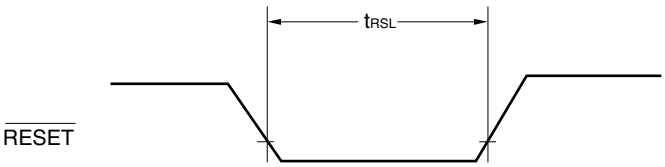
Interrupt Request Input Timing



Key Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing



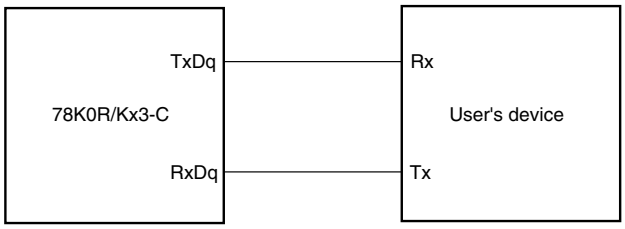
(2) Serial interface: Serial array unit (1/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

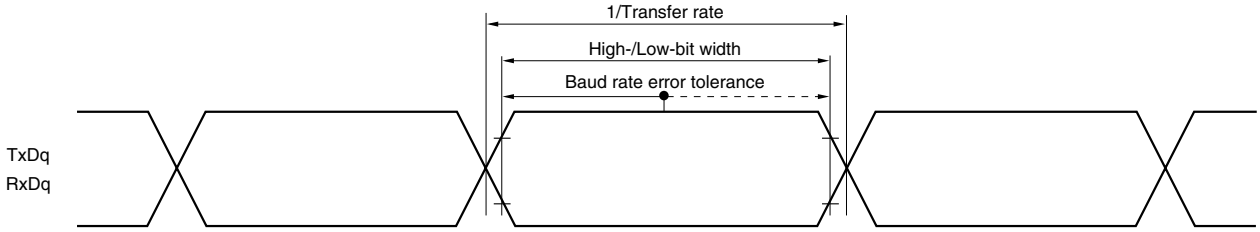
(a) During communication at same potential (UART mode) (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					f_{MCK}/6	bps
		f_{CLK} = 20 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 RxDq and the normal output mode for TxDq by using the PIMg and POMg registers.

Remarks 1. q: UART number (q = 0 to 2), g: PIM and POM number (g = 0, 1, 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. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

(2) Serial interface: Serial array unit (2/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

(b) During communication at same potential (CSI mode) (master mode, $\overline{SCKp}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V	200 Note 1			ns
		2.7 V ≤ V_{DD} < 4.0 V	300 Note 1			ns
$\overline{SCKp}$ high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V	t_{KCY1}/2 – 20			ns
		2.7 V ≤ V_{DD} < 4.0 V	t_{KCY1}/2 – 35			ns
SIp setup time (to $\overline{SCKp}$↑) Note 2	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V	70			ns
		2.7 V ≤ V_{DD} < 4.0 V	100			ns
SIp hold time (from $\overline{SCKp}$↑) Note 3	t_{KSI1}		30			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 4	t_{KSO1}	C = 30 pF Note 5			40	ns

Notes 1. Set t_{KCY1} in $4/f_{CLK} \le t_{KCY1}$.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

5. C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

Caution Select the normal input buffer for SIp and the normal output mode for SOp and $\overline{SCKp}$ by using the PIMg and POMg registers.

Remarks 1. p: CSI number (p = 00, 01, 10, 20), g: PIM and POM number (g = 0, 1, 14)

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

3. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

(2) Serial interface: Serial array unit (3/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

(c) During communication at same potential (CSI mode) (slave mode, $\overline{SCKp}$... external clock input)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY2}	4.0 V ≤ V_{DD} = EV_{DD}		6/f_{MCK}			ns
		2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V	16 MHz < f_{MCK}	8/f_{MCK}			ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}			ns
$\overline{SCKp}$ high-/low-level width	t_{KH2}, t_{KL2}			f_{KCY2}/2			ns
SIp setup time (to $\overline{SCKp}$↑) Note 1	t_{SIK2}			80			ns
SIp hold time (from $\overline{SCKp}$↑) Note 2	t_{KSI2}			1/f_{MCK}+50			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	4.0 V ≤ V_{DD} ≤ 5.5 V			2/f_{MCK}+45	ns
			2.7 V ≤ V_{DD} < 4.0 V			2/f_{MCK}+57	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SOp output line.

Caution Select the normal input buffer for SIp and $\overline{SCKp}$ and the normal output mode for SOp by using the PIMg and POMg registers.

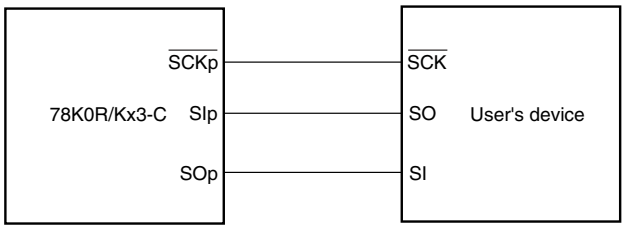
Remarks 1. p: CSI number (p = 00, 01, 10, 20), g: PIM and POM number (g = 0, 1, 14)

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 2))

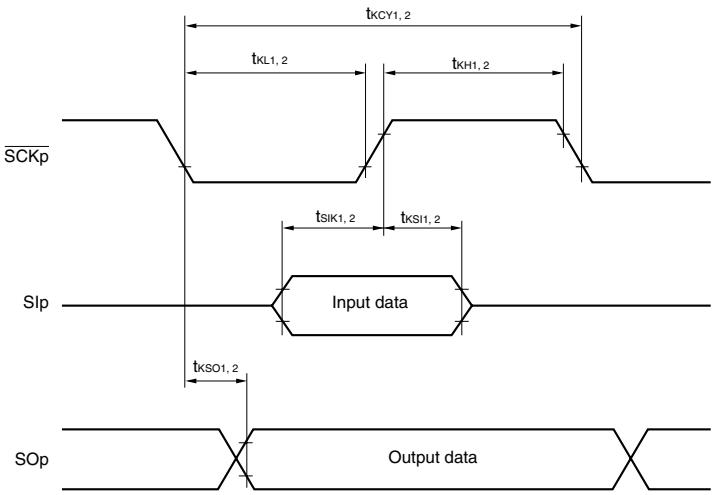
3. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

(2) Serial interface: Serial array unit (4/17)

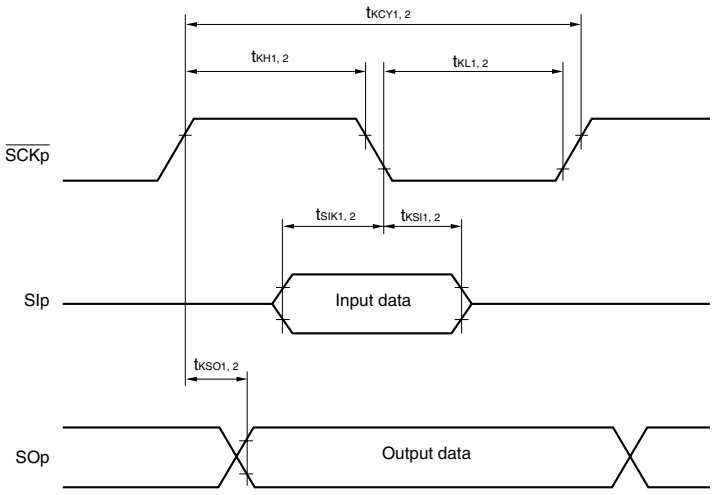
CSI mode connection diagram (during communication at same potential)



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, 20)

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

(2) Serial interface: Serial array unit (5/17)

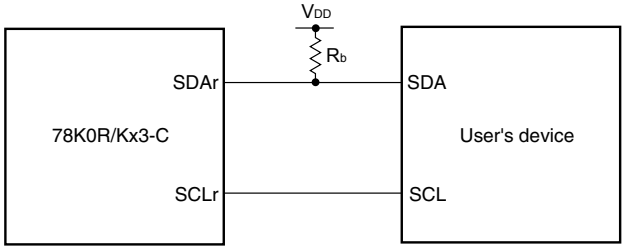
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

(d) During communication at same potentia (simplified I^2C mode)

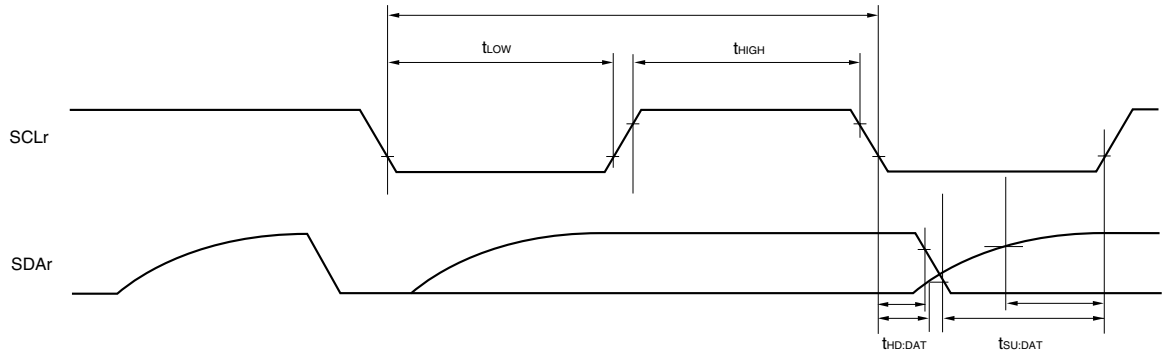
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		400 Note	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1200		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1200		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1/f_{MCK}+120		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	660	ns

Note Set f_{SCL} in f_{SCL} ≤ f_{MCK}/4.

Simplified I^2C mode mode connection diagram (during communication at same potential)



Simplified I^2C 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[Ω]:Communication line (SDAr) pull-up resistance,
C_b[F]: Communication line (SCLr, SDAr) load capacitance

2. r: IIC number (r = 10, 20), 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, 2), mn = 02, 10)

4. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

(2) Serial interface: Serial array unit (6/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

(e) 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 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V				f_{MCK}/6	bps
				f_{CLK} = 20 MHz, f_{MCK} = f_{CLK}			3.3	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V				f_{MCK}/6	bps
				f_{CLK} = 20 MHz, f_{MCK} = f_{CLK}			3.3	Mbps

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. V_b[V]: Communication line voltage

2. q: UART number (q = 0 to 2), g: PIM and POM number (g = 0, 1, 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 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

5. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

(2) Serial interface: Serial array unit (7/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

(e) 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 ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V				**Note 1**	
				f_{CLK} = 16.8 MHz, f_{MCK} = f_{CLK}, C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V			2.8 Note 2	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V				**Note 3**	
				f_{CLK} = 19.2 MHz, f_{MCK} = f_{CLK}, C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V			1.2 Note 4	Mbps

Notes 1. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* 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_{MCK}/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

4. This value as an example is calculated when the conditions described in the “Conditions” column are met. 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_{DD} tolerance) mode for TxDq by using the PIMg and POMg registers.

(Remarks are given on the next page.)

(2) Serial interface: Serial array unit (8/17)

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 = 0 to 2), g: PIM and POM number (g = 0, 1, 14)

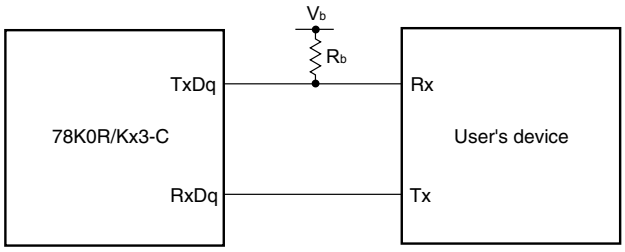
3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3))

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.
$4.0\ V \le V_{DD} \le 5.5\ V$, $2.7\ V \le V_b \le 4.0\ V$: $V_{IH} = 2.2\ V$, $V_{IL} = 0.8\ V$
$2.7\ V \le V_{DD} \le 4.0\ V$, $2.3\ V \le V_b \le 2.7\ V$: $V_{IH} = 2.0\ V$, $V_{IL} = 0.5\ V$

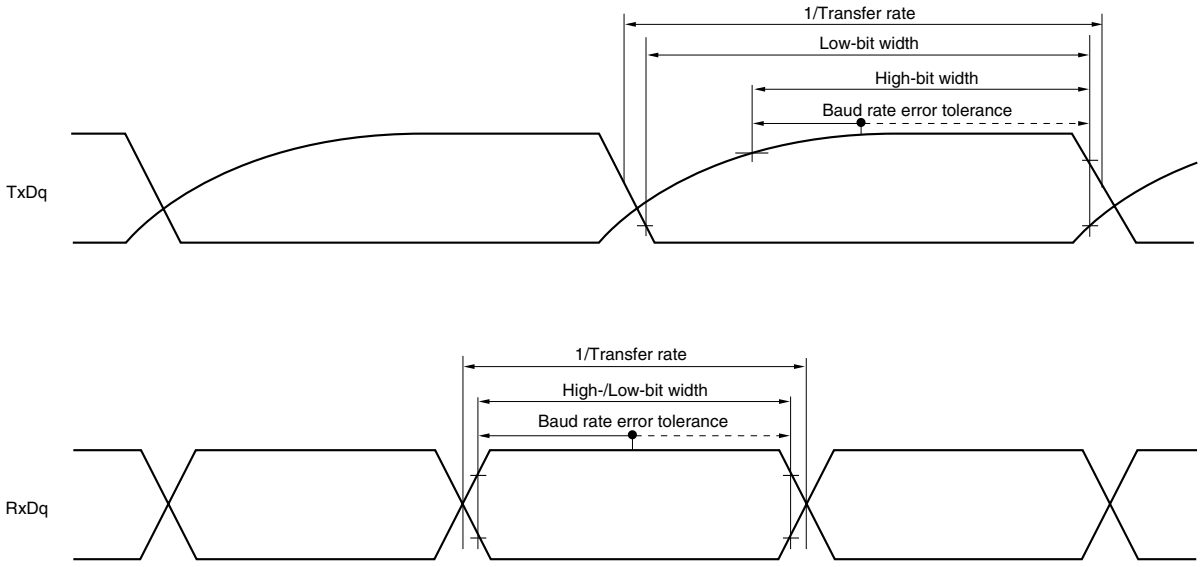
5. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

(2) Serial interface: Serial array unit (9/17)

UART mode connection diagram (communication at different potential)



UART mode bit width (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[Ω]:Communication line (TxDq) pull-up resistance, V_b[V]: Communication line voltage

2. q: UART number (q = 0 to 2), g: PIM and POM number (g = 0, 1, 14)

(2) Serial interface: Serial array unit (10/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

(f) Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{SCKp}$... internal clock output) (1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	400 Note 1			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	800 Note 1			ns
$\overline{SCKp}$ high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}/2 – 75			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 170			ns
$\overline{SCKp}$ low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	t_{KCY1}/2 – 20			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 35			ns
SIp setup time (to $\overline{SCKp}$↑) Note 2	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	150			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	275			ns
SIp hold time (from $\overline{SCKp}$↑) Note 2	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	30			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 2	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			120	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			215	ns

Notes 1. Set t_{KCY1} in $4/f_{CLK} \le t_{KCY1}$.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Caution Select the TTL input buffer for SIp 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, 20), g: PIM and POM number (g = 0, 1, 14)

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

3. R_b[Ω]:Communication line ($\overline{SCKp}$, SOp) pull-up resistance,
C_b[F]: Communication line ($\overline{SCKp}$, SOp) load capacitance, V_b[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 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

5. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

(2) Serial interface: Serial array unit (11/17)

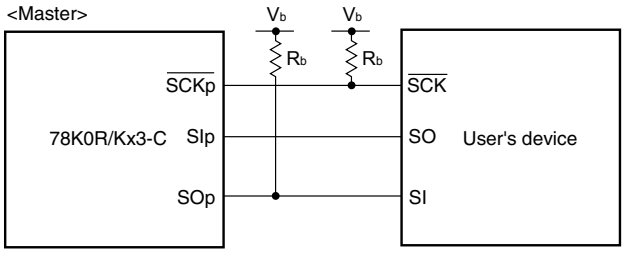
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

(f) Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{SCKp}$... internal clock output) (2/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SIp setup time (to $\overline{SCKp}$↓) Note	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	70			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	100			ns
SIp hold time (from $\overline{SCKp}$↓) Note	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	30			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{SCKp}$↑ to SOp output Note	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			40	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			40	ns

Note When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

CSI mode connection diagram (communication at different potential)



Caution Select the TTL input buffer for SIp 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, 20), g: PIM and POM number (g = 0, 1, 14)

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

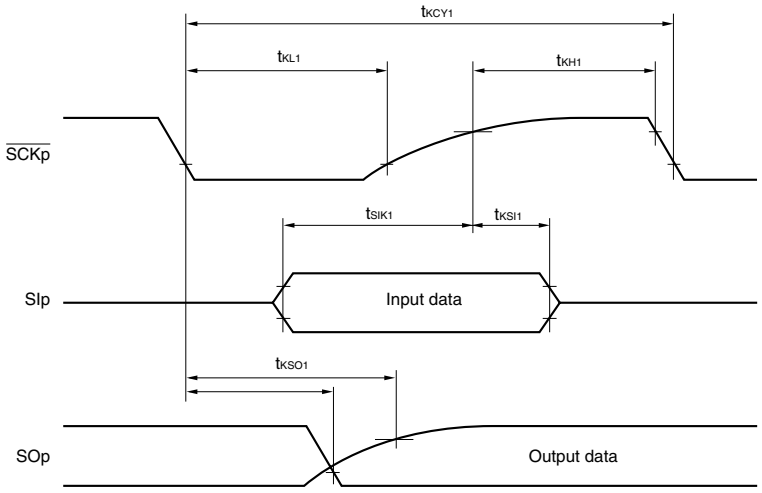
3. R_b[Ω]:Communication line ($\overline{SCKp}$, SOp) pull-up resistance,
C_b[F]: Communication line ($\overline{SCKp}$, SOp) load capacitance, V_b[V]: Communication line voltage

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 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

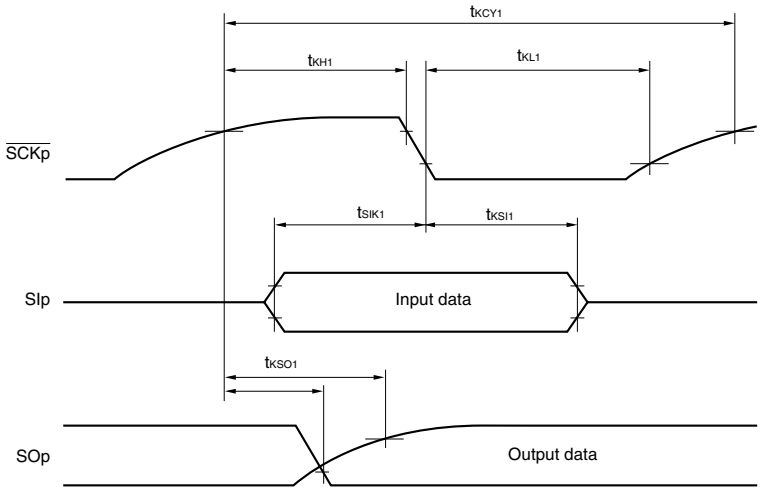
5. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

(2) Serial interface: Serial array unit (12/17)

CSI mode serial transfer timing (master mode) (communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (master mode) (communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Caution Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and $\overline{\text{SCKp}}$ by using the PIMg and POMg registers.

Remarks 1. p: CSI number (p = 01, 10, 20), g: PIM and POM number (g = 0, 1, 14)

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

(2) Serial interface: Serial array unit (13/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

(g) Communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, $\overline{SCKp}$... external clock input)

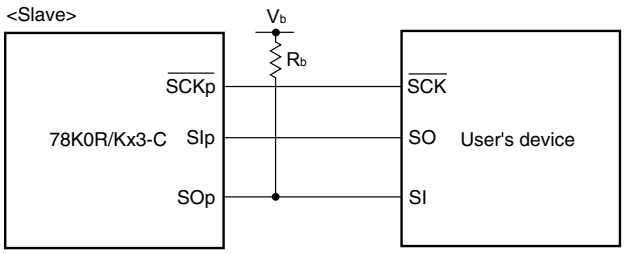
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	13.6 MHz < f_{MCK}	10/f_{MCK}			ns
			6.8 MHz < f_{MCK} ≤ 13.6 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 6.8 MHz	6/f_{MCK}			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	18.5 MHz < f_{MCK}	16/f_{MCK}			ns
			14.8 MHz < f_{MCK} ≤ 18.5 MHz	14/f_{MCK}			ns
			11.1 MHz < f_{MCK} ≤ 14.8 MHz	12/f_{MCK}			ns
			7.4 MHz < f_{MCK} ≤ 11.1 MHz	10/f_{MCK}			ns
			3.7 MHz < f_{MCK} ≤ 7.4 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 3.7 MHz	6/f_{MCK}			ns
$\overline{SCKp}$ high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		f_{KCY2}/2 – 20			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{KCY2}/2 – 35			ns
SIp setup time (to $\overline{SCKp}$↑) Note 1	t_{SIK2}			90			ns
SIp hold time (from $\overline{SCKp}$↑) Note 2	t_{KSI2}			1/f_{MCK} + 50			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 3	t_{KSO2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ				2/f_{MCK} + 120	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ				2/f_{MCK} + 230	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

CSI mode connection diagram (communication at different potential)



(Caution and Remark are given on the next page.)

(2) Serial interface: Serial array unit (14/17)

Caution Select the TTL input buffer for SIp and $\overline{SCK}$p 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, 20), g: PIM and POM number (g = 0, 1, 14)

2. R_b[Ω]:Communication line (SOp) pull-up resistance,
C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

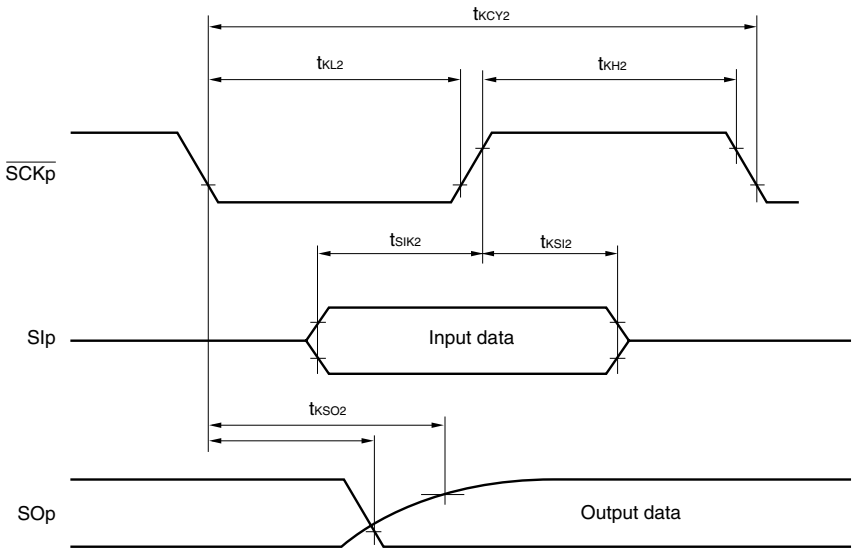
3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2))

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 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

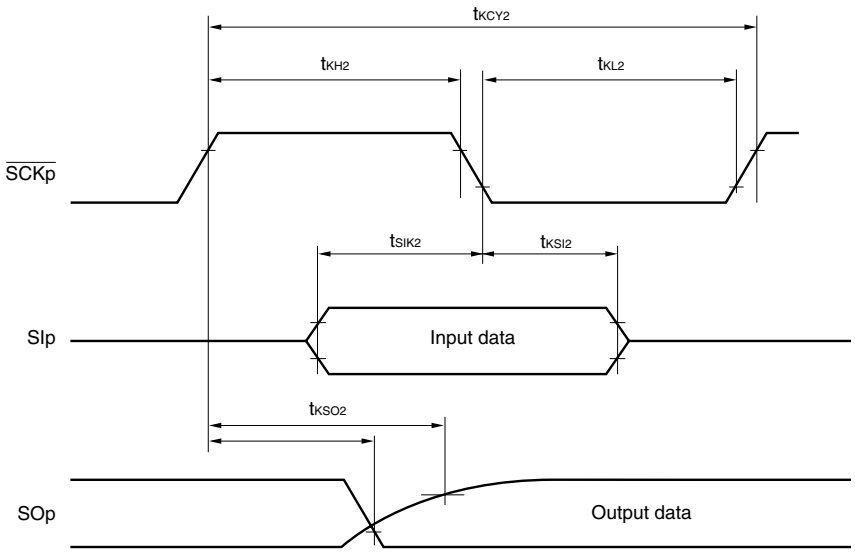
5. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

(2) Serial interface: Serial array unit (15/17)

CSI mode serial transfer timing (slave mode) (communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (slave mode) (communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Caution Select the TTL input buffer for SIp and $\overline{\text{SCKp}}$ 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, 20), g: PIM and POM number (g = 0, 1, 14)

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

(2) Serial interface: Serial array unit (16/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

(h) Communication at different potential (2.5 V, 3 V) (simplified I^2C mode)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ		400 Note	kHz
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		400 Note	kHz
Hold time when SCLr = “L”	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	1275		ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1275		ns
Hold time when SCLr = “H”	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	655		ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	655		ns
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	1/f_{MCK} + 190		ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1/f_{MCK} + 190		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 1.4 kΩ	0	640	ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	660	ns

Note Set f_{SCL} in f_{SCL} ≤ f_{MCK}/4.

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[Ω]:Communication line (SDAr, SCLr) pull-up resistance,
C_b[F]: Communication line (SDAr, SCLr) load capacitance, V_b[V]: Communication line voltage

2. r: IIC number (r = 10, 20), g: PIM, POM number (g = 0, 14)

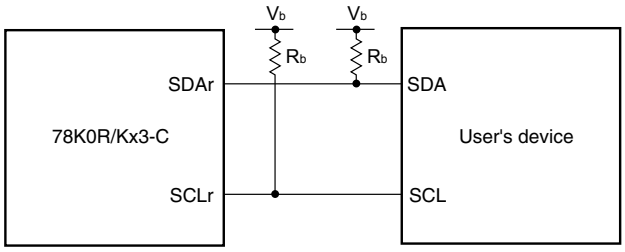
3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10)

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 simplified I^2C mode mode.
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

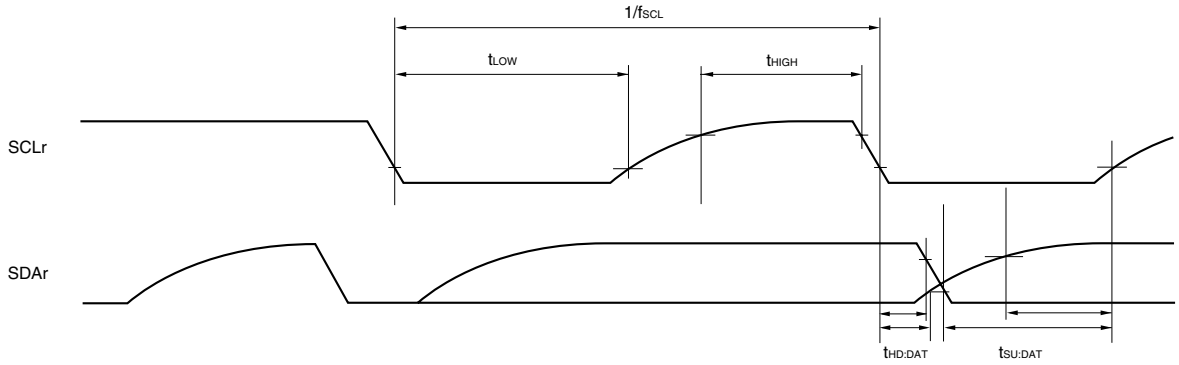
5. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

(2) Serial interface: Serial array unit (17/17)

Simplified I^2C mode connection diagram (communication at different potential)



Simplified I^2C mode serial transfer timing (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[Ω]:Communication line (SDAr, SCLr) pull-up resistance, V_b[V]: Communication line voltage

2. r: IIC number (r = 10, 20), g: PIM and POM number (g = 0, 14)

(3) Serial interface: IICA

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

(a) IICA

Parameter	Symbol	Conditions	Standard Mode		Fast Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCL0 clock frequency	f_{SCL}	Fast mode: 3.5 MHz ≤ f_{CLK} Standard mode: 1 MHz ≤ f_{CLK}	0	100	0	400	kHz
Setup time of restart condition[Note 1]	$t_{SU:STA}$		4.7		0.6		μs
Hold time	$t_{HD:STA}$		4.0		0.6		μs
Hold time when SCL0 = “L”	t_{LOW}		4.7		1.3		μs
Hold time when SCL0 = “H”	t_{HIGH}		4.0		0.6		μs
Data setup time (reception)	$t_{SU:DAT}$		250		100		ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0	3.45	0	0.9	μs
Setup time of stop condition	$t_{SU:STO}$		4.0		0.6		μs
Bus-free time	t_{BUF}		4.7		1.3		μs

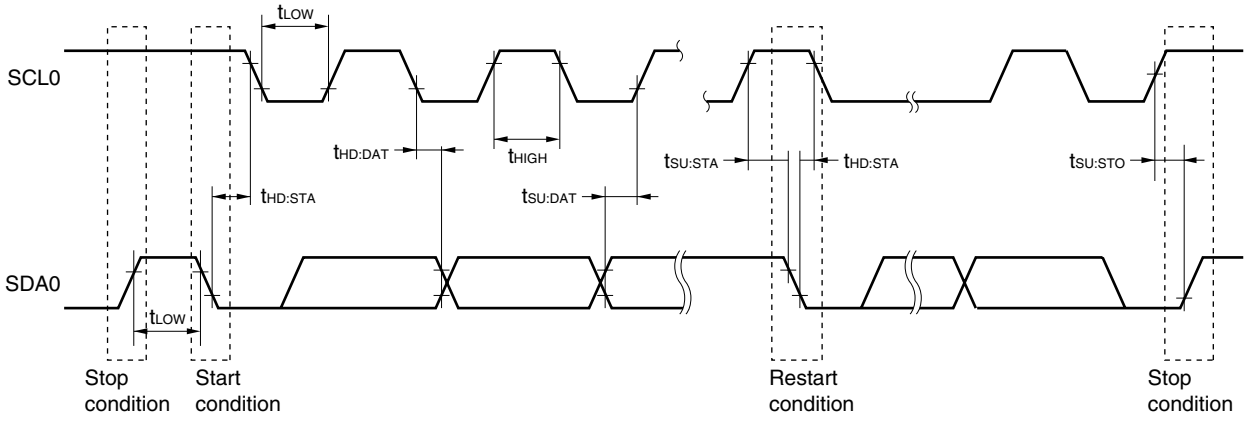
Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a wait state is inserted in the $\overline{ACK}$ (acknowledge) timing.

Remarks 1. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

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

IICA serial transfer timing



(4) Serial interface: On-chip debug (UART)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 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			3.33	Mbps
TOOL1 output frequency	f_{TOOL1}	2.7 V ≤ V_{DD} ≤ 5.5 V			10	MHz

<R> **CEC Transmission/Reception Circuit**

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 3.63 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input hysteresis width	IHYS	P62	V_{DD} = 3.3 V±10%		0.4		V
Rise time of CECIO	t_R	P62	CECIO mode[Note] C_b = 1600 pF, R_b = 27 kΩ			250	μs
			CECIO mode[Note] C_b = 7700 pF, R_b = 3 kΩ			250	μs
Fall time of CECIO	t_F	P62	CECIO mode[Note] C_b = 1600 pF, R_b = 27 kΩ			50	μs
			CECIO mode[Note] C_b = 7700 pF, R_b = 3 kΩ			50	μs

Note When a pull-up resistor and a diode are connected. (PF62 = 1, PU62 = 1)

Remarks 1. C_b[F]: Communication line load capacitance, R_b[Ω]:Communication line pull-up resistance

2. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

A/D Converter Characteristics

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}					10	bit
Overall error[Notes 1, 2]	AINL					±0.35	%FSR
Conversion time	t_{CONV}	4.0 V ≤ AV_{REF} ≤ 5.5 V	High speed mode 1	2.5		66.6	μs
			Normal mode	5.2		66.6	μs
		2.7 V ≤ AV_{REF} < 4.0 V	High speed mode 2	3.5		66.6	μs
			Normal mode	8.6		66.6	μs
Zero-scale error[Notes 1, 2]	EZS					±0.25	%FSR
Full-scale error[Notes 1, 2]	EFS					±0.25	%FSR
Integral linearity error[Note 1]	ILE					±2.5	LSB
Differential linearity error[Note 1]	DLE					±1.5	LSB
Analog input voltage	V_{AIN}			AV_{SS}		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

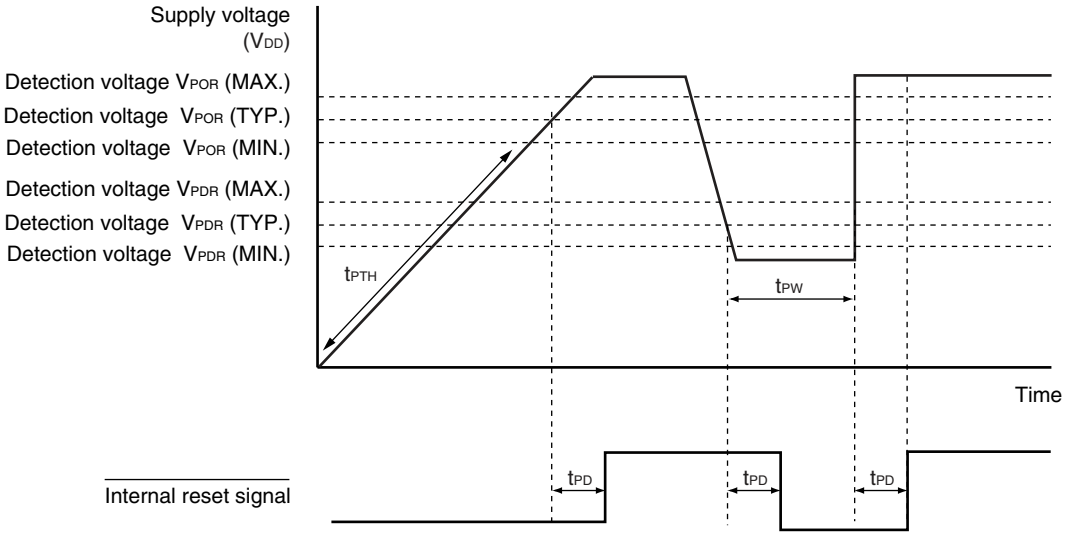
2. This value is indicated as a ratio (%FSR) to the full-scale value.

Remark For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

POC Circuit Characteristics (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.52	1.61	1.70	V
	V_{PDR}	Power supply fall time	1.5	1.59	1.68	V
Power supply voltage rise inclination	t_{PTH}	Change inclination of V_{DD}: 0 V → V_{POR}	0.5			V/ms
Minimum pulse width	t_{PW}	When the voltage drops	200			μs
Detection delay time	t_{PD}				200	μs

POC Circuit Timing



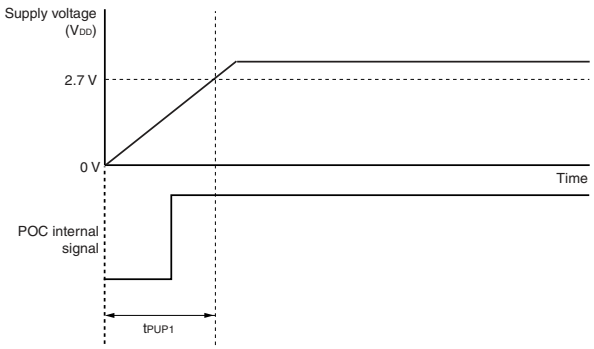
Supply Voltage Rise Time (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 2.7 V (V_{DD} (MIN.))[Note] (V_{DD}: 0 V → 2.7 V)	t_{PUP1}	LVI default start function stopped is set (LVIOFF (Option Byte) = 1), when $\overline{\text{RESET}}$ input is not used			3.6	ms
Maximum time to rise to 2.7 V (V_{DD} (MIN.))[Note] (releasing $\overline{\text{RESET}}$ input → V_{DD}: 2.7 V)	t_{PUP2}	LVI default start function stopped is set (LVIOFF (Option Byte) = 1), when $\overline{\text{RESET}}$ input is used			1.88	ms

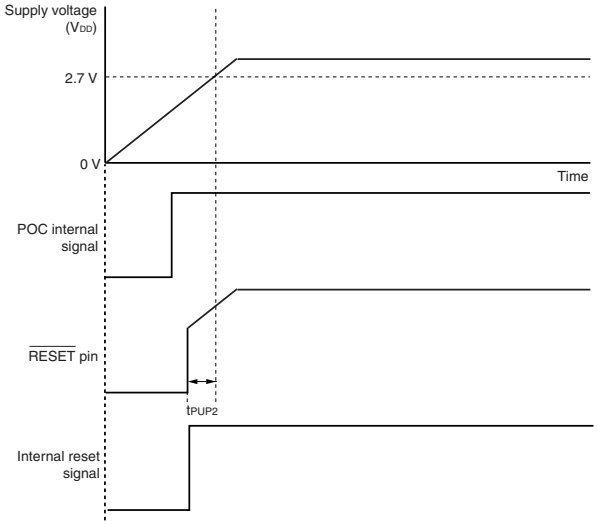
Note Make sure to raise the power supply in a shorter time than this.

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°C, $V_{PDR} \le V_{DD} = EV_{DD0} = EV_{DD1} \le 5.5$ V, 2.7 V ≤ $AV_{REF} \le V_{DD}$, $V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS}$ = 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
	External input pin[Note 1]	V_{EXLVI}	EXLVI < V_{DD}, 2.7 V ≤ V_{DD} ≤ 5.5 V	1.11	1.21	1.31	V
	Power supply voltage on power application	V_{PUPLVI}	When LVI default start function enabled is set	1.87	2.07	2.27	V
Minimum pulse width		t_{LW}		200			μs
Detection delay time		t_D				200	μs
Operation stabilization wait time[Note 2]		t_{LWAIT}				10	μs

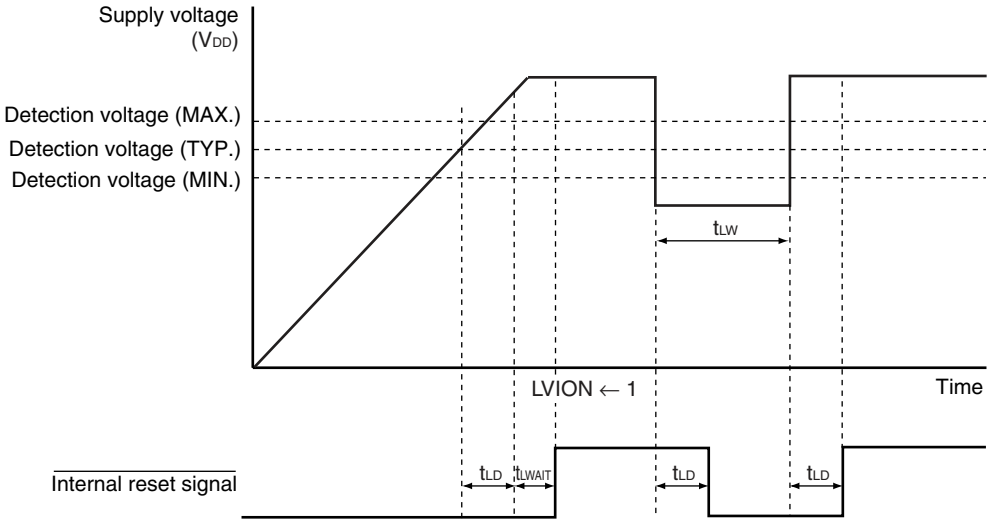
Notes 1. The EXLVI/P120/INTP0 pin is used.

2. Time required from setting bit 7 (LVION) of the low-voltage detection register (LVIM) to 1 to operation stabilization

Remarks 1. $V_{LVI(n-1)} > V_{LVIn}$: n = 1 to 9

2. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

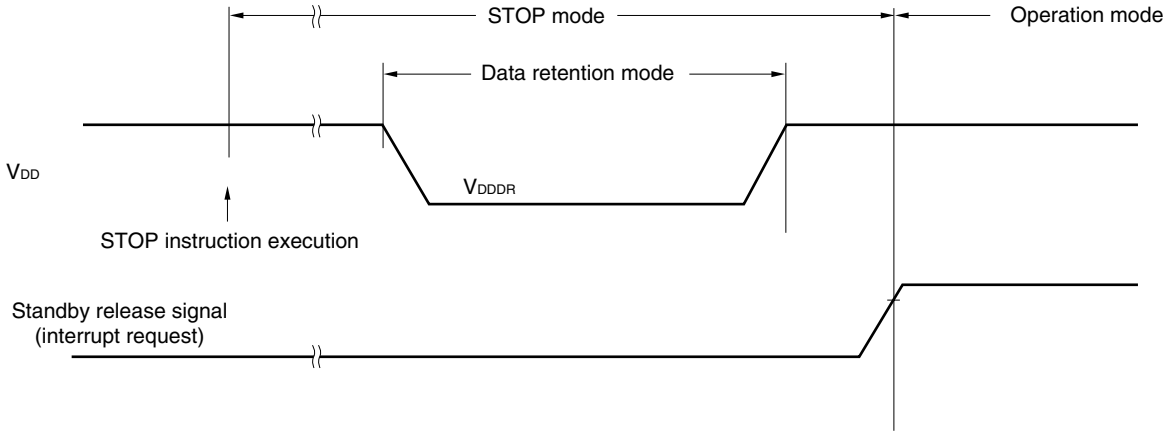
LVI Circuit Timing



Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.5 Note		5.5	V

Note The value depends on the POC detection voltage. When the voltage drops, the data is retained before 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°C Note, 2.7 V ≤ V_{DD} = EV_{DD0} = EV_{DD1} ≤ 5.5 V, 2.7 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS0} = EV_{SS1} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
V_{DD} supply current	I_{DD}	Typ. = 10 MHz, Max. = 20 MHz			6	20	mA
CPU/peripheral hardware clock frequency	f_{CLK}			2		20	MHz
Number of rewrites (number of deletes per block)	C_{erwr}	Used for updating programs When using flash memory programmer and Renesas Electronics self programming library	Retained for 15 years	1,000			Times
		Used for updating data When using Renesas Electronics EEPROM emulation library	Retained for 5 years	10,000			Times

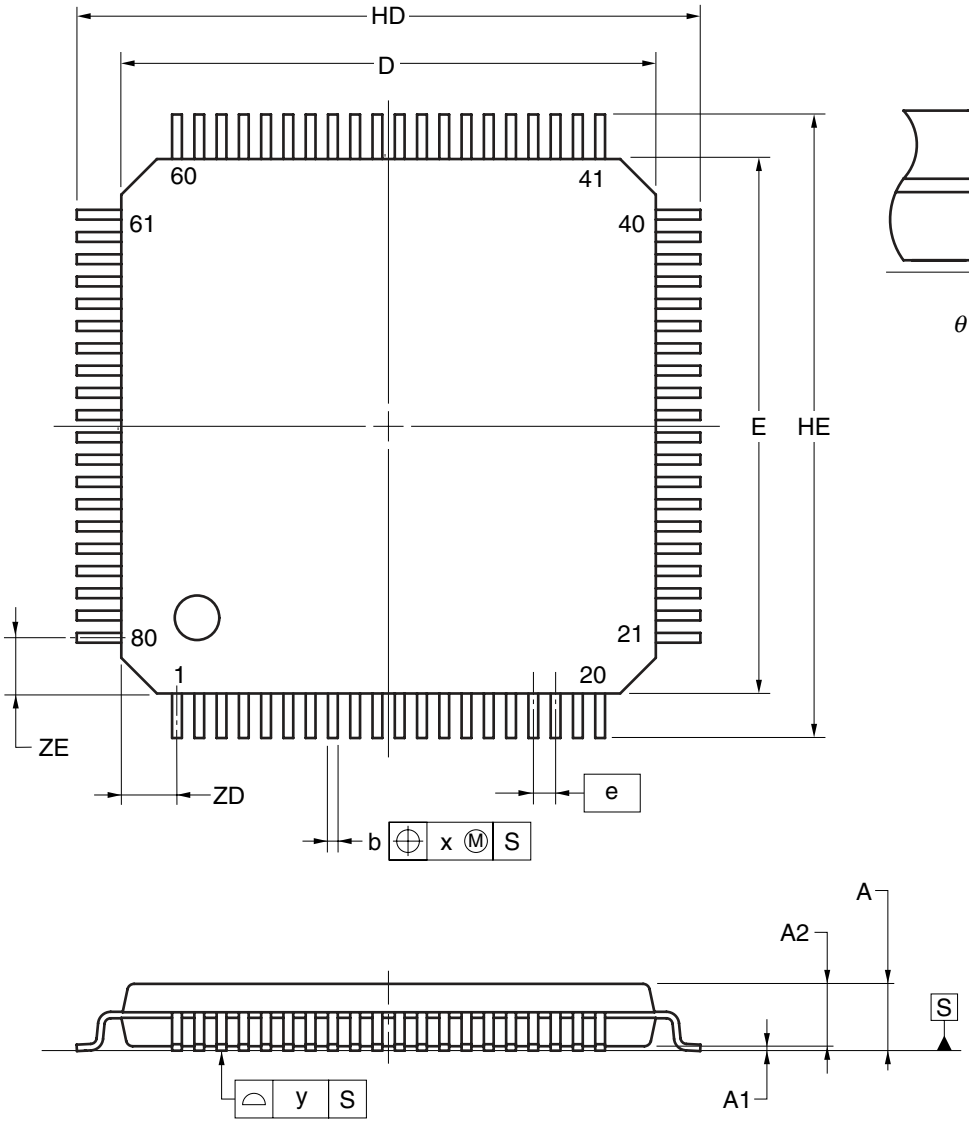
Note When data is written by the flash memory programmer: T_A = 0 to +50°C

Remarks 1. When updating data multiple times, use the flash memory as one for updating data.

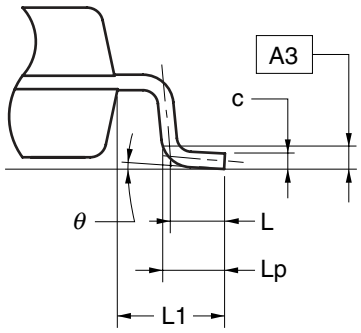
2. For the 78K0R/KF3-C, read EV_{DD0} and EV_{DD1} as EV_{DD} and EV_{SS0} and EV_{SS1} as EV_{SS}.

CHAPTER 30 PACKAGE DRAWINGS

80-PIN PLASTIC LQFP (FINE PITCH) (12x12)



detail of lead end



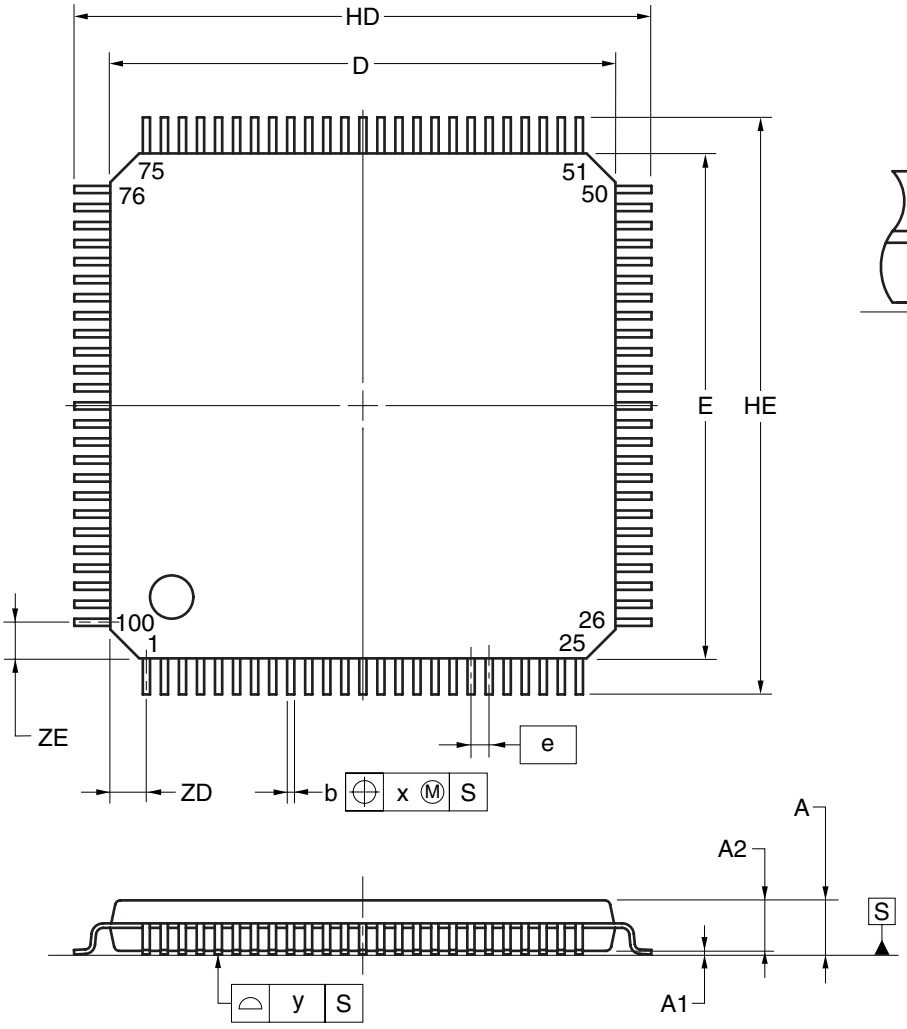
(UNIT:mm)

ITEM	DIMENSIONS
D	12.00±0.20
E	12.00±0.20
HD	14.00±0.20
HE	14.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

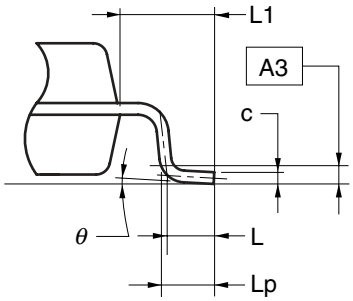
P80GK-50-GAK

NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

100-PIN PLASTIC LQFP (FINE PITCH) (14x14)



detail of lead end



(UNIT:mm)

ITEM	DIMENSIONS
D	14.00±0.20
E	14.00±0.20
HD	16.00±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	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.00
ZE	1.00

P100GC-50-UEU-1

CHAPTER 31 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, please contact a Renesas Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www2.renesas.com/pkg/en/mount/index.html)

Table 31-1. Surface Mounting Type Soldering Conditions

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[Note] (after that, prebake at 125°C for 10 to 72 hours)	IR60-107-3

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Cautions 1. Do not use different soldering methods together (except for partial heating).

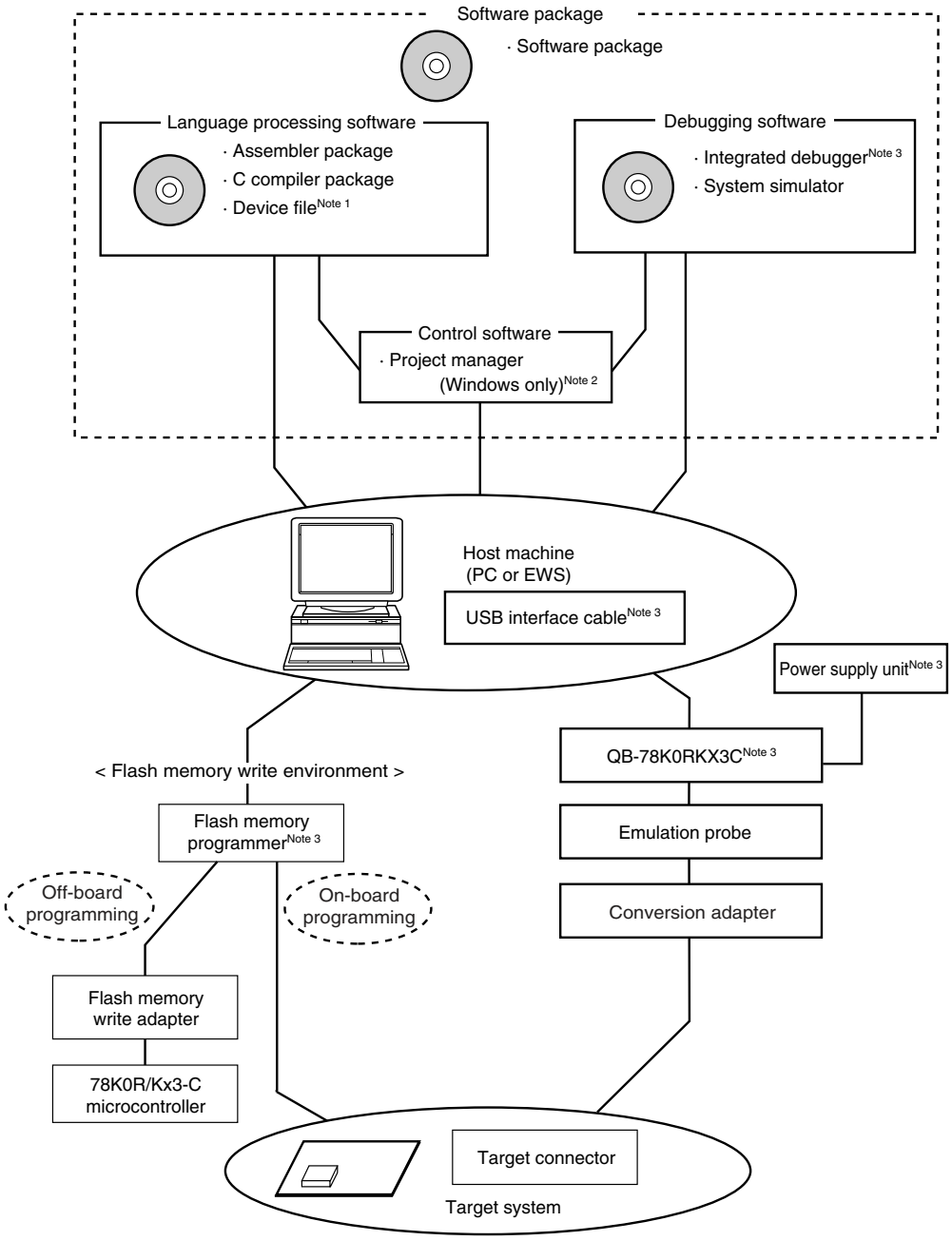
2. The 78K0R/Kx3-C microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the 78K0R/Kx3-C.
Figure A-1 shows the development tool configuration.

Figure A-1. Development Tool Configuration (1/2)

(1) When using the in-circuit emulator QB-78K0RKX3C



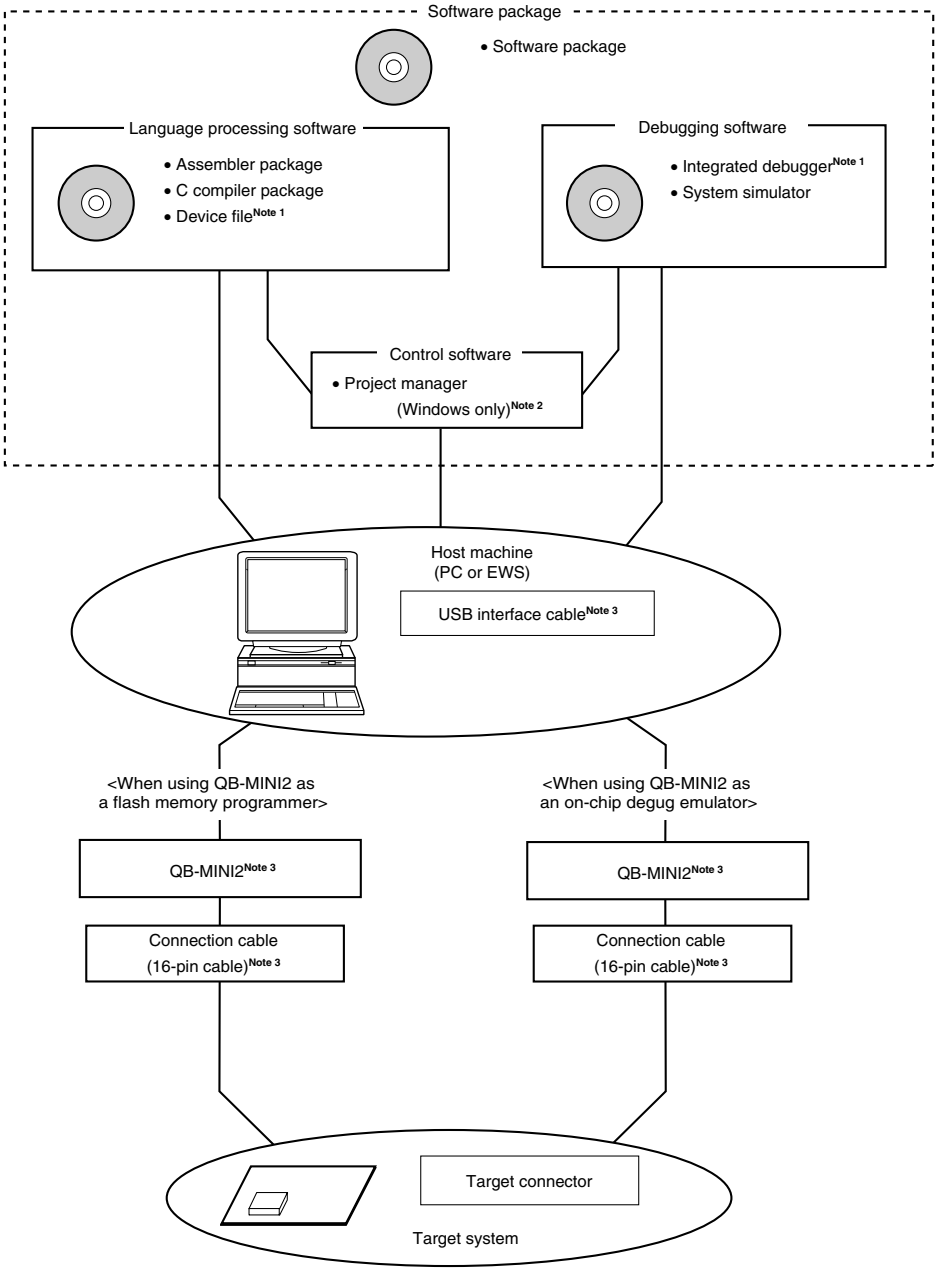
Notes 1. Download the device file for the 78K0R/Kx3-C (DF781849) from the download site for development tools (http://www2.renesas.com/micro/en/ods/index.html).

2. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

3. In-circuit emulator QB-78K0RKX3C 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 the 78K0R/Kx3-C (DF781849) and the integrated debugger (ID78K0R-QB) from the download site for development tools (http://www2.renesas.com/micro/en/ods/index.html).

2. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

3. 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://www2.renesas.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: μS××××SP78K0R

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SP78K0R

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		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 (DF781849[Note]). **<Precaution when using RA78K0R in PC environment>** 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: μS××××RA78K0R
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). **<Precaution when using CC78K0R in PC environment>** 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: μS××××CC78K0R
DF781849[Note] 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: μS××××DF781849

Note The DF781849 can be used in common with the RA78K0R, CC78K0R, SM+ for 78K0R, and ID78K0R-QB. Download the DF781849 from the download site for development tools (http://www2.renesas.com/micro/en/ods/index.html).

Remark ××××in the part number differs depending on the host machine and OS used.

μS××××RA78K0R
μS××××CC78K0R

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

μS××××DF781849

××××	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. **<Caution>** 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 programmer PG-FP5 and FL-PR5

PG-FP5 and FL-PR5 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
FA-xxxx[Note] Flash memory programming adapter	Flash memory programming adapter used connected to the flash memory programmer for use.

Note The part numbers of the flash memory programming adapter and the packages of the target device are described below.

Package		Flash Memory Programming Adapter
78K0R/KF3-C	80-pin plastic LQFP (GK-GAK type)	FA-78F1847GK-GAK-RX
78K0R/KG3-C	100-pin plastic LQFP (GC-UEU type)	FA-78F1849GC-UEU-RX

Remarks 1. The FL-PR5 and FA-xxxx are products of Naito Densei Machida Mfg. Co., Ltd.

2. Use the latest version of the flash memory programming adapter.

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/Kx3-C microcontrollers. 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 the 78K0R/Kx3-C, 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://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html).

A.5 Debugging Tools (Hardware)

A.5.1 When using in-circuit emulator

QB-78K0RKX3C[Note] In-circuit emulator	This in-circuit emulator serves to debug hardware and software when developing application systems using the 78K0R/KF3-C, 78K0R/KG3-C microcontrollers. 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-xxxx-EA-xxx[Note] Exchange adapter	This exchange adapter is used to perform pin conversion from the in-circuit emulator to target connector.
QB-xxxx-YS-xxxx[Note] Space adapter	This space adapter is used to adjust the height between the target system and in-circuit emulator.
QB-xxxx-YQ-xxx[Note] YQ connector	This YQ connector is used to connect the target connector and exchange adapter.
QB-xxxx-HQ-xxx[Note] Mount adapter	This mount adapter is used to mount the target device with socket.
QB-xxxx-NQ-xxx[Note] Target connector	This target connector is used to mount on the target system.

Note The part numbers of the exchange adapter, space adapter, YQ connector, mount adapter, and target connector and the packages of the target device are described below.

Package		Exchange Adapter	Space Adapter	YQ Connector	Mount Adapter	Target Connector
78K0R/KF3-C	80-pin plastic LQFP (GK-GAK type)	QB-80GK-EA-09T	QB-80GK-YS-01T	QB-80GK-YQ-01T	QB-80GK-HQ-01T	QB-80GK-NQ-01T
78K0R/KG3-C	100-pin plastic LQFP (GC-UEU type)	QB-100GC-EA-07T	QB-100GC-YS-01T	QB-100GC-YQ-01T	QB-100GC-HQ-01T	QB-100GC-NQ-01T

Remarks 1. The QB-78K0RKX3C 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-78K0RKX3C-ZZZ	QB-78K0RKX3C	None			
QB-78K0RKX3C-T80GK		QB-144-EP-02S	QB-80GK-EA-09T	QB-80GK-YQ-01T	QB-80GK-NQ-01T
QB-78K0RKX3C-T100GC			QB-100GC-EA-07T	QB-100GC-YQ-01T	QB-100GC-NQ-01T

A.5.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This on-chip debug emulator serves to debug hardware and software when developing application systems using the 78K0R/Kx3-C 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 the 78K0R/Kx3-C, 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://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html).

A.6 Debugging Tools (Software)

SM+ for 78K0R System simulator	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 (DF781849).
	Part number: μS××××SM781000
ID78K0R-QB Integrated debugger	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 (DF781849).
	Part number: μS××××ID78K0R-QB

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SM781000
μS××××ID78K0R-QB

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

APPENDIX B REVISION HISTORY

B.1 Major Revisions in This Edition

Page	Description	Classification
CHAPTER 4 PORT FUNCTION		
p. 169	Addition of **Figure 4-55. Relationship Between PF6 Register and PU6 Register**	(c)
p. 170	Addition of **Figure 4-57. Relationship Between PF6 Register and PU6 Register**	(c)
CHAPTER 12 SERIAL INTERFACES IICA		
p. 539	Change of **Figure 12-7. Format of IICA Status Register (IICS) (3/3)**	(c)
CHAPTER 13 CEC TRANSMISSION/RECEPTION CIRCUIT		
p. 610, 611	Addition of description to **13.1 Functions of CEC Transmission/Reception Circuit**	(c)
CHAPTER 29 ELECTRICAL SPECIFICATIONS		
p. 943	Change of **CEC Transmission/Reception circuit Characteristics**	(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

B.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

Edition	Description	Chapter
1st edition	Change of status of μPD78F1846A, 78F1847A, 78F1848A, and 78F1849A from under development to mass production	Throughout
	Change of status of 80-pin plastic LQFP (fine pitch) (12 × 12) of 78K0R/KF3 and 100-pin plastic LQFP (fine pitch) (14 × 14) of 78K0R/KG3 from under development to mass production	
	Deletion of target from the capacitance value of the capacitor connected to the REGC pin	
	Change URL of Renesas Electronics website	
	Change of **Figure 21-3. Example of Software Processing After Reset Release (1/2)**	CHAPTER 21 POWER-ON-CLEAR CIRCUIT
	Change of **Figure 22-11. Example of Software Processing After Reset Release (1/2)**	CHAPTER 22 LOW-VOLTAGE DETECTOR
	Change of specifications from target specifications to formal specifications	CHAPTER 29 ELECTRICAL SPECIFICATIONS
	Change of internal high-speed oscillation clock frequency of **Internal Oscillator Characteristics**	
	Addition of **Recommended oscillator circuit constants**	
	Addition of chapter	CHAPTER 31 RECOMMENDED SOLDERING CONDITIONS

78K0R/Kx3-C User's Manual: Hardware

Publication Date: Rev.0.01 Mar 1, 2010
Rev.2.00 Sep 30, 2011

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1

78K0R/Kx3-C

Renesas Electronics Corporation

R01UH0072EJ0200