NEC

User's Manual

78K0R/KE3

16-bit Single-Chip Microcontrollers

μPD78F1142
μPD78F1143
μPD78F1144
μPD78F1145
μPD78F1146

The 78K0R/KE3 has an on-chip debug function.
Do not use this product for mass production because its reliability cannot be guaranteed after the on-chip debug function has been used, due to issues with respect to the number of times the flash memory can be rewritten. NEC Electronics does not accept complaints concerning this product after the on-chip debug function has been used.

Document No. U17854EJ6V0UD00 (6th edition)
Date Published August 2007 NS

© NEC Electronics Corporation 2006
Printed in Japan

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

② HANDLING OF UNUSED INPUT PINS

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

③ PRECAUTION AGAINST ESD

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

④ STATUS BEFORE INITIALIZATION

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

⑤ POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current.

The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

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

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

EEPROM is a trademark of NEC Electronics Corporation.

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

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

- **The information in this document is current as of August, 2007. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

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

  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.

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

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

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

(Note)

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

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

M8E 02. 11-1

INTRODUCTION

Readers

This manual is intended for user engineers who wish to understand the functions of the 78K0R/KE3 and design and develop application systems and programs for these devices.
The target products are as follows.

78K0R/KE3: μPD78F1142, 78F1143, 78F1144, 78F1145, 78F1146

Purpose

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

Organization

The 78K0R/KE3 manual is separated into two parts: this manual and the instructions edition (common to the 78K0R Microcontroller Series).

78K0R/KE3 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 Series instructions:
  → Refer to the separate document **78K0R Microcontroller Instructions User's Manual (U17792E)**.

Conventions

Data significance:	Higher digits on the left and lower digits on the right	
Active low representations:	$\overline{\times\times\times}$ (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/KE3 User's Manual	This manual
78K0R Microcontroller Instructions User's Manual	U17792E
78K0R Microcontroller Self Programming Library Type01 User's Manual[Note]	U18706E

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

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

Document Name		Document No.
CC78K0R Ver. 1.00 C Compiler	Operation	U17838E
	Language	U17837E
RA78K0R Ver. 1.00 Assembler Package	Operation	U17836E
	Language	U17835E
SM+ System Simulator	Operation	U18010E
PM+ Ver. 6.20		U17990E
ID78K0R-QB Ver. 3.20 Integrated Debugger	Operation	U17839E

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

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

Documents Related to Flash Memory Programming

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

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

Other Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE − Products and Packages −	X13769X
Semiconductor Device Mount Manual	**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.necel.com/pkg/en/mount/index.html).

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

CONTENTS

CHAPTER 1 OUTLINE 16

1.1 Features 16
1.2 Applications 17
1.3 Ordering Information 17
1.4 Pin Configuration (Top View) 18
1.5 78K0R Microcontroller Lineup 21
1.6 Block Diagram 22
1.7 Outline of Functions 23

CHAPTER 2 PIN FUNCTIONS 25

2.1 Pin Function List 25
2.2 Description of Pin Functions 30
2.2.1 P00 to P06 (port 0) 30
2.2.2 P10 to P17 (port 1) 31
2.2.3 P20 to P27 (port 2) 32
2.2.4 P30, P31 (port 3) 32
2.2.5 P40 to P43 (port 4) 33
2.2.6 P50 to P55 (port 5) 34
2.2.7 P60 to P63 (port 6) 34
2.2.8 P70 to P77 (port 7) 34
2.2.9 P120 to P124 (port 12) 35
2.2.10 P130 (port 13) 36
2.2.11 P140, P141 (port 14) 36
2.2.12 AV_{REF} 36
2.2.13 AV_{SS} 36
2.2.14 $\overline{RESET}$ 36
2.2.15 REGC 37
2.2.16 V_{DD}, EV_{DD} 37
2.2.17 V_{SS}, EV_{SS} 37
2.2.18 FLMD0 37
2.3 Pin I/O Circuits and Recommended Connection of Unused Pins 38

CHAPTER 3 CPU ARCHITECTURE 42

3.1 Memory Space 42
3.1.1 Internal program memory space 49
3.1.2 Mirror area 51
3.1.3 Internal data memory space 52
3.1.4 Special function register (SFR) area 53
3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area 53
3.1.6 Data memory addressing 54
3.2 Processor Registers 59
3.2.1 Control registers 59
3.2.2 General-purpose registers 61

3.2.3 ES and CS registers 63
3.2.4 Special function registers (SFRs) 64
3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers) 70
3.3 Instruction Address Addressing 75
3.3.1 Relative addressing 75
3.3.2 Immediate addressing 75
3.3.3 Table indirect addressing 76
3.3.4 Register direct addressing 77
3.4 Addressing for Processing Data Addresses 78
3.4.1 Implied addressing 78
3.4.2 Register addressing 78
3.4.3 Direct addressing 79
3.4.4 Short direct addressing 80
3.4.5 SFR addressing 81
3.4.6 Register indirect addressing 82
3.4.7 Based addressing 83
3.4.8 Based indexed addressing 86
3.4.9 Stack addressing 87

CHAPTER 4 PORT FUNCTIONS 88

4.1 Port Functions 88
4.2 Port Configuration 91
4.2.1 Port 0 92
4.2.2 Port 1 98
4.2.3 Port 2 104
4.2.4 Port 3 106
4.2.5 Port 4 107
4.2.6 Port 5 112
4.2.7 Port 6 114
4.2.8 Port 7 116
4.2.9 Port 12 117
4.2.10 Port 13 120
4.2.11 Port 14 121
4.3 Registers Controlling Port Function 123
4.4 Port Function Operations 130
4.4.1 Writing to I/O port 130
4.4.2 Reading from I/O port 130
4.4.3 Operations on I/O port 130
4.4.4 Connecting to external device with different power potential (2.5V, 3 V) 131
4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function 133
4.6 Cautions on 1-Bit Manipulation Instruction for Port Register n (Pn) 135

CHAPTER 5 CLOCK GENERATOR 136

5.1 Functions of Clock Generator 136
5.2 Configuration of Clock Generator 137
5.3 Registers Controlling Clock Generator 139
5.4 System Clock Oscillator 153
5.4.1 X1 oscillator 153

5.4.2 XT1 oscillator 153
5.4.3 Internal high-speed oscillator 156
5.4.4 Internal low-speed oscillator 156
5.4.5 Prescaler 156
5.5 Clock Generator Operation 157
5.6 Controlling Clock 161
5.6.1 Example of controlling high-speed system clock 161
5.6.2 Example of controlling internal high-speed oscillation clock 164
5.6.3 Example of controlling subsystem clock 166
5.6.4 Example of controlling internal low-speed oscillation clock 168
5.6.5 CPU clock status transition diagram 169
5.6.6 Condition before changing CPU clock and processing after changing CPU clock 174
5.6.7 Time required for switchover of CPU clock and main system clock 175
5.6.8 Conditions before clock oscillation is stopped 176

CHAPTER 6 TIMER ARRAY UNIT 177

6.1 Functions of Timer Array Unit 177
6.1.1 Functions of each channel when it operates independently 177
6.1.2 Functions of each channel when it operates with another channel 178
6.1.3 LIN-bus supporting function (channel 7 only) 178
6.2 Configuration of Timer Array Unit 179
6.3 Registers Controlling Timer Array Unit 184
6.4 Channel Output (TO0n pin) Control 205
6.4.1 TO0n pin output circuit configuration 205
6.4.2 TO0n Pin Output Setting 206
6.4.3 Cautions on Channel Output Operation 206
6.4.4 Collective manipulation of TO0n bits 210
6.4.5 Timer Interrupt and TO0n Pin Output at Operation Start 211
6.5 Channel Input (TI0n Pin) Control 212
6.5.1 TI0n edge detection circuit 212
6.6 Basic Function of Timer Array Unit 213
6.6.1 Overview of single-operation function and combination-operation function 213
6.6.2 Basic rules of combination-operation function 213
6.6.3 Applicable range of basic rules of combination-operation function 214
6.7 Operation of Timer Array Unit as Independent Channel 215
6.7.1 Operation as interval timer/square wave output 215
6.7.2 Operation as external event counter 219
6.7.3 Operation as frequency divider 222
6.7.4 Operation as input pulse interval measurement 226
6.7.5 Operation as input signal high-/low-level width measurement 230
6.8 Operation of Plural Channels of Timer Array Unit 234
6.8.1 Operation as PWM function 234
6.8.2 Operation as one-shot pulse output function 241
6.8.3 Operation as multiple PWM output function 248

CHAPTER 7 REAL-TIME COUNTER 255

7.1 Functions of Real-Time Counter 255
7.2 Configuration of Real-Time Counter 255

7.3 Registers Controlling Real-Time Counter 257
7.4 Real-Time Counter Operation 269
7.4.1 Starting operation of real-time counter 269
7.4.2 Reading/writing real-time counter 270
7.4.3 Setting alarm of real-time counter 272

CHAPTER 8 WATCHDOG TIMER 273

8.1 Functions of Watchdog Timer 273
8.2 Configuration of Watchdog Timer 274
8.3 Register Controlling Watchdog Timer 275
8.4 Operation of Watchdog Timer 276
8.4.1 Controlling operation of watchdog timer 276
8.4.2 Setting overflow time of watchdog timer 277
8.4.3 Setting window open period of watchdog timer 278
8.4.4 Setting watchdog timer interval interrupt 279

CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER 280

9.1 Functions of Clock Output/Buzzer Output Controller 280
9.2 Configuration of Clock Output/Buzzer Output Controller 281
9.3 Registers Controlling Clock Output/Buzzer Output Controller 281
9.4 Operations of Clock Output/Buzzer Output Controller 283
9.4.1 Operation as output pin 283

CHAPTER 10 A/D CONVERTER 284

10.1 Function of A/D Converter 284
10.2 Configuration of A/D Converter 285
10.3 Registers Used in A/D Converter 287
10.4 A/D Converter Operations 296
10.4.1 Basic operations of A/D converter 296
10.4.2 Input voltage and conversion results 298
10.4.3 A/D converter operation mode 299
10.5 How to Read A/D Converter Characteristics Table 301
10.6 Cautions for A/D Converter 303

CHAPTER 11 SERIAL ARRAY UNIT 307

11.1 Functions of Serial Array Unit 307
11.1.1 3-wire serial I/O (CSI00, CSI10) 307
11.1.2 UART (UART0, UART1, UART3) 308
11.1.3 Simplified I^2C (IIC10) 308
11.2 Configuration of Serial Array Unit 309
11.3 Registers Controlling Serial Array Unit 314
11.4 Operation stop mode 336
11.4.1 Stopping the operation by units 336
11.4.2 Stopping the operation by channels 337
11.5 Operation of 3-Wire Serial I/O (CSI00, CSI10) Communication 339
11.5.1 Master transmission 340
11.5.2 Master reception 349

11.5.3 Master transmission/reception 355
11.5.4 Slave transmission 363
11.5.5 Slave reception 372
11.5.6 Slave transmission/reception 378
11.5.7 Calculating transfer clock frequency 387
11.6 Operation of UART (UART0, UART1, UART3) Communication 389
11.6.1 UART transmission 390
11.6.2 UART reception 400
11.6.3 LIN transmission 407
11.6.4 LIN reception 410
11.6.5 Calculating baud rate 415
11.7 Operation of Simplified I^2C (IIC10) Communication 419
11.7.1 Address field transmission 420
11.7.2 Data transmission 425
11.7.3 Data reception 428
11.7.4 Stop condition generation 431
11.7.5 Calculating transfer rate 432
11.8 Processing Procedure in Case of Error 435
11.9 Relationship Between Register Settings and Pins 437

CHAPTER 12 SERIAL INTERFACE IIC0 442

12.1 Functions of Serial Interface IIC0 442
12.2 Configuration of Serial Interface IIC0 445
12.3 Registers to Controlling Serial Interface IIC0 448
12.4 I^2C Bus Mode Functions 460
12.4.1 Pin configuration 460
12.5 I^2C Bus Definitions and Control Methods 461
12.5.1 Start conditions 461
12.5.2 Addresses 462
12.5.3 Transfer direction specification 462
12.5.4 Transfer clock setting method 463
12.5.5 Acknowledge ($\overline{ACK}$) 464
12.5.6 Stop condition 466
12.5.7 Wait 467
12.5.8 Canceling wait 469
12.5.9 Interrupt request (INTIIC0) generation timing and wait control 470
12.5.10 Address match detection method 471
12.5.11 Error detection 471
12.5.12 Extension code 471
12.5.13 Arbitration 472
12.5.14 Wakeup function 473
12.5.15 Communication reservation 474
12.5.16 Cautions 478
12.5.17 Communication operations 479
12.5.18 Timing of I^2C interrupt request (INTIIC0) occurrence 487
12.6 Timing Charts 508

CHAPTER 13 MULTIPLIER 515

13.1 Functions of Multiplier 515
13.2 Configuration of Multiplier 516
13.3 Operation of Multiplier 517

CHAPTER 14 DMA CONTROLLER 518

14.1 Functions of DMA Controller 518
14.2 Configuration of DMA Controller 519
14.3 Registers to Controlling DMA Controller 522
14.4 Operation of DMA Controller 525
14.4.1 Operation procedure 525
14.4.2 Transfer mode 526
14.4.3 Termination of DMA transfer 526
14.5 Example of Setting of DMA Controller 527
14.5.1 CSI consecutive transmission 527
14.5.2 Consecutive capturing of A/D conversion results 529
14.5.3 UART consecutive reception + ACK transmission 531
14.5.4 Holding DMA transfer pending by DWAITn 533
14.5.5 Forced termination by software 534
14.6 Cautions on Using DMA Controller 535

CHAPTER 15 INTERRUPT FUNCTIONS 537

15.1 Interrupt Function Types 537
15.2 Interrupt Sources and Configuration 538
15.3 Registers Controlling Interrupt Functions 541
15.4 Interrupt Servicing Operations 551
15.4.1 Maskable interrupt acknowledgment 551
15.4.2 Software interrupt request acknowledgment 553
15.4.3 Multiple interrupt servicing 554
15.4.4 Interrupt request hold 557

CHAPTER 16 KEY INTERRUPT FUNCTION 558

16.1 Functions of Key Interrupt 558
16.2 Configuration of Key Interrupt 558
16.3 Register Controlling Key Interrupt 559

CHAPTER 17 STANDBY FUNCTION 560

17.1 Standby Function and Configuration 560
17.1.1 Standby function 560
17.1.2 Registers controlling standby function 560
17.2 Standby Function Operation 563
17.2.1 HALT mode 563
17.2.2 STOP mode 568

CHAPTER 18 RESET FUNCTION 574

18.1 Register for Confirming Reset Source 582

CHAPTER 19 POWER-ON-CLEAR CIRCUIT 583

19.1 Functions of Power-on-Clear Circuit 583
19.2 Configuration of Power-on-Clear Circuit 584
19.3 Operation of Power-on-Clear Circuit 584
19.4 Cautions for Power-on-Clear Circuit 587

CHAPTER 20 LOW-VOLTAGE DETECTOR 589

20.1 Functions of Low-Voltage Detector 589
20.2 Configuration of Low-Voltage Detector 590
20.3 Registers Controlling Low-Voltage Detector 590
20.4 Operation of Low-Voltage Detector 595
20.4.1 When used as reset 596
20.4.2 When used as interrupt 602
20.5 Cautions for Low-Voltage Detector 608

CHAPTER 21 REGULATOR 612

21.1 Regulator Overview 612
21.2 Registers Controlling Regulator 612

CHAPTER 22 OPTION BYTE 614

22.1 Functions of Option Bytes 614
22.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H) 614
22.1.2 On-chip debug option byte (000C3H/ 010C3H) 615
22.2 Format of User Option Byte 615
22.3 Format of On-chip Debug Option Byte 617
22.4 Setting of Option Byte 617

CHAPTER 23 FLASH MEMORY 618

23.1 Writing with Flash Memory Programmer 618
23.2 Programming Environment 620
23.3 Communication Mode 620
23.4 Connection of Pins on Board 621
23.4.1 FLMD0 pin 621
23.4.2 TOOL0 pin 622
23.4.3 $\overline{\text{RESET}}$ pin 622
23.4.4 Port pins 623
23.4.5 REGC pin 623
23.4.6 X1 and X2 pins 623
23.4.7 Power supply 623
23.5 Registers that Control Flash Memory 623
23.6 Programming Method 624
23.6.1 Controlling flash memory 624

23.6.2 Flash memory programming mode 624
23.6.3 Selecting communication mode 625
23.6.4 Communication commands....625
23.7 Security Settings 627
23.8 Flash Memory Programming by Self-Programming 629
23.8.1 Boot swap function....631
23.8.2 Flash shield window function 633

CHAPTER 24 ON-CHIP DEBUG FUNCTION 634

24.1 Connecting QB-MINI2 to 78K0R/KE3 634
24.2 On-Chip Debug Security ID 635
24.3 Securing of user resources 635

CHAPTER 25 BCD CORRECTION CIRCUIT 637

25.1 BCD Correction Circuit Function 637
25.2 Registers Used by BCD Correction Circuit 637
25.3 BCD Correction Circuit Operation 638

CHAPTER 26 INSTRUCTION SET 640

26.1 Conventions Used in Operation List 640
26.1.1 Operand identifiers and specification methods 640
26.1.2 Description of operation column....641
26.1.3 Description of flag operation column....642
26.1.4 PREFIX Instruction 642
26.2 Operation List 643

CHAPTER 27 ELECTRICAL SPECIFICATIONS 660

CHAPTER 28 PACKAGE DRAWINGS 704

APPENDIX A DEVELOPMENT TOOLS 708

A.1 Software Package 711
A.2 Language Processing Software 711
A.3 Control Software 712
A.4 Flash Memory Programming Tools 712
A.4.1 When using flash memory programmer FG-FP5, FL-PR5, FG-FP4, and FL-PR4....712
A.4.2 When using on-chip debug emulator with programming function QB-MINI2 713
A.5 Debugging Tools (Hardware) 713
A.5.1 When using in-circuit emulator QB-78K0RKX3 713
A.5.2 When using on-chip debug emulator with programming function QB-MINI2 714
A.6 Debugging Tools (Software) 715

APPENDIX B REVISION HISTORY 716

B.1 Major Revisions in This Edition 716
B.2 Revision History of Preceding Editions 720

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

Item / Part Number	Program Memory (ROM)		Data Memory (RAM)
μPD78F1142	Flash memory	64 KB	4 KB
μPD78F1143		96 KB	6 KB
μPD78F1144		128 KB	8 KB
μPD78F1145		192 KB	10 KB
μPD78F1146		256 KB	12 KB

- On-chip single-power-supply flash memory (with prohibition of chip erase/block erase/writing function)
- Self-programming (with boot swap function/flash shield window function)
- On-chip debug function
- On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)
- On-chip watchdog timer (operable with the on-chip internal low-speed oscillation clock)
- On-chip multiplier (16 bits × 16 bits)
- On-chip key interrupt function
- On-chip clock output/buzzer output controller
- On-chip BCD adjustment
- I/O ports: 55 (N-ch open drain: 4)
- Timer: 10 channels
  - 16-bit timer: 8 channels
  - Watchdog timer: 1 channel
  - Real-time counter: 1 channel
- Serial interface
  - UART/CSI: 1 channel
  - UART/CSI/simplified I^2C: 1 channel
  - UART (LIN-bus supported): 1 channel
  - I^2C: 1 channel
- 10-bit resolution A/D converter (AV_{REF} = 2.3 to 5.5 V): 8 channels
- Power supply voltage: V_{DD} = 1.8 to 5.5 V
- Operating ambient temperature: T_A = −40 to +85°C

1.2 Applications

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

1.3 Ordering Information

- **Flash memory version (lead-free products)**

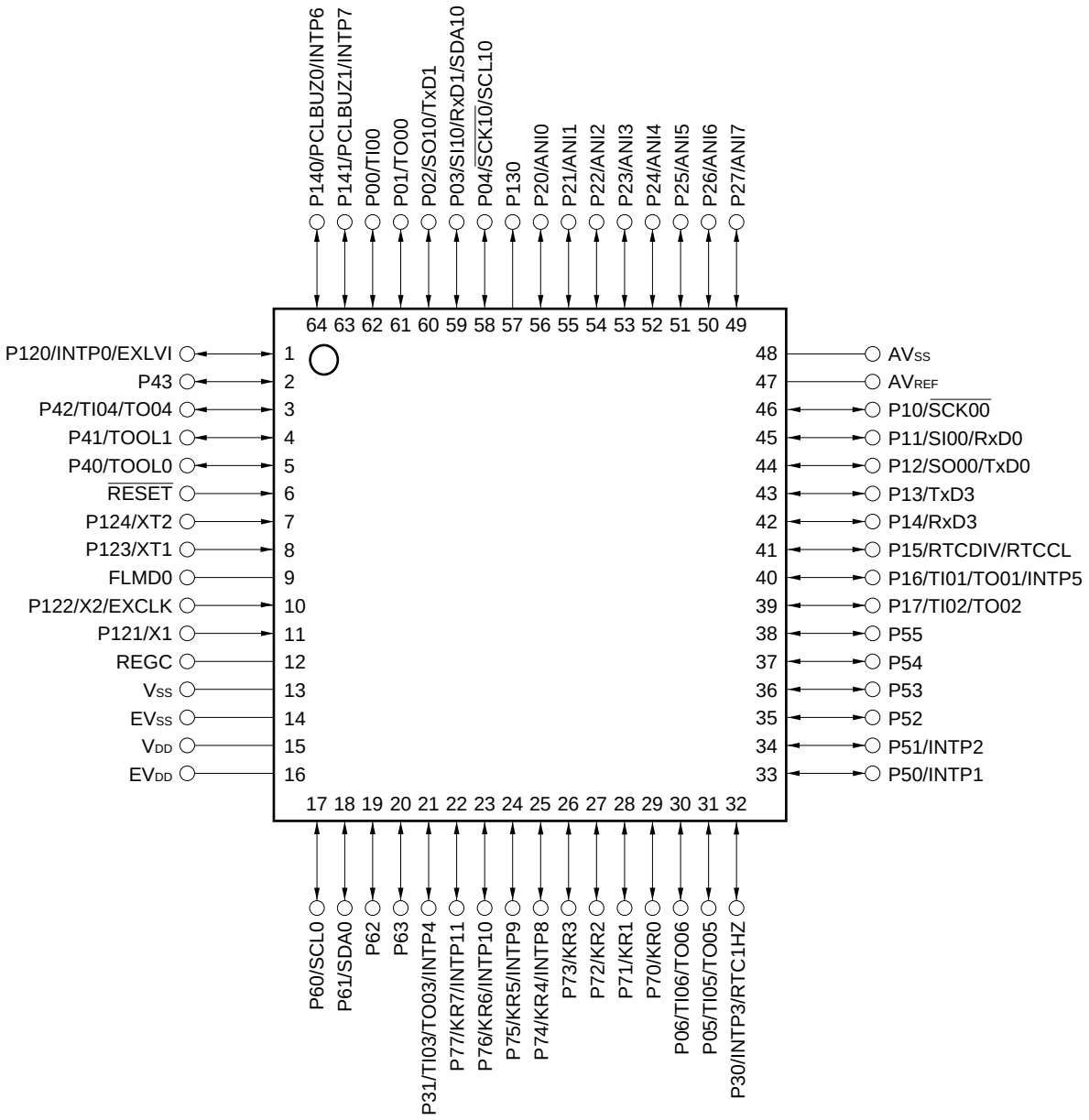
	Part Number	Package
	μPD78F1142GK-GAJ-AX	64-pin plastic LQFP (12 × 12)
	μPD78F1143GK-GAJ-AX	64-pin plastic LQFP (12 × 12)
	μPD78F1144GK-GAJ-AX	64-pin plastic LQFP (12 × 12)
	μPD78F1145GK-GAJ-AX	64-pin plastic LQFP (12 × 12)
	μPD78F1146GK-GAJ-AX	64-pin plastic LQFP (12 × 12)
	μPD78F1142GB-GAH-AX	64-pin plastic LQFP (fine pitch) (10 × 10)
	μPD78F1143GB-GAH-AX	64-pin plastic LQFP (fine pitch) (10 × 10)
	μPD78F1144GB-GAH-AX	64-pin plastic LQFP (fine pitch) (10 × 10)
	μPD78F1145GB-GAH-AX	64-pin plastic LQFP (fine pitch) (10 × 10)
	μPD78F1146GB-GAH-AX	64-pin plastic LQFP (fine pitch) (10 × 10)
<R>	μPD78F1142GA-HAB-AX[Note]	64-pin plastic TQFP (fine pitch) (7 × 7)
<R>	μPD78F1143GA-HAB-AX[Note]	64-pin plastic TQFP (fine pitch) (7 × 7)
<R>	μPD78F1144GA-HAB-AX[Note]	64-pin plastic TQFP (fine pitch) (7 × 7)
<R>	μPD78F1145GA-HAB-AX[Note]	64-pin plastic TQFP (fine pitch) (7 × 7)
<R>	μPD78F1146GA-HAB-AX[Note]	64-pin plastic TQFP (fine pitch) (7 × 7)
<R>	μPD78F1142F1-AN1-A[Note]	64-pin plastic FBGA (5 × 5)
<R>	μPD78F1143F1-AN1-A[Note]	64-pin plastic FBGA (5 × 5)
<R>	μPD78F1144F1-AN1-A[Note]	64-pin plastic FBGA (5 × 5)
<R>	μPD78F1145F1-AN1-A[Note]	64-pin plastic FBGA (5 × 5)
<R>	μPD78F1146F1-AN1-A[Note]	64-pin plastic FBGA (5 × 5)

Note Under development

Caution The 78K0R/KE3 has an on-chip debug function. Do not use this product for mass production, because its reliability cannot be guaranteed after the on-chip debug function has been used, with respect to the number of times the flash memory can be rewritten. NEC Electronics does not accept complaints about this product after the on-chip debug function has been used.

1.4 Pin Configuration (Top View)

- 64-pin plastic LQFP (12 × 12)
- 64-pin plastic LQFP (fine pitch) (10 × 10)
- <R> 64-pin plastic TQFP (fine pitch) (7 × 7) [Note]



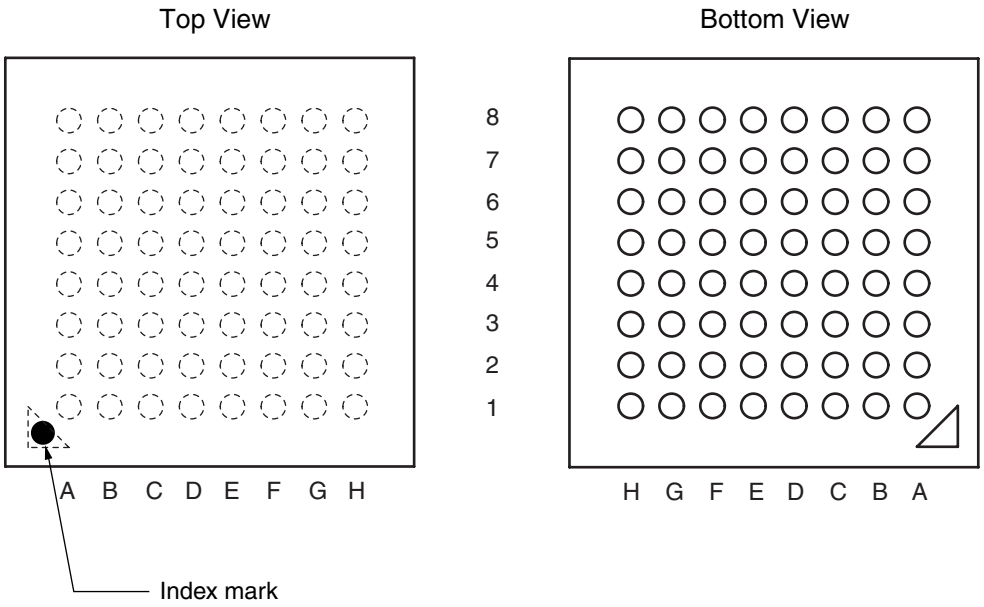
Note Under development

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

2. Make EVDD the same potential as VDD.

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

<R> • 64-pin plastic FBGA (5 × 5) [Note]



Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
A1	P30/INTP3/RTC1HZ	C1	P17/TI02/TO02	E1	P13/TxD3	G1	AV_{REF}
A2	P05/TI05/TO05	C2	P10/$\overline{SCK00}$	E2	P15/RTCDIV/RTCCL	G2	P24/ANI4
A3	P06/TI06/TO06	C3	P53	E3	P54	G3	P23/ANI3
A4	P74/KR4/INTP8	C4	P70/KR0	E4	P52	G4	P22/ANI2
A5	P76/KR6/INTP10	C5	P63	E5	P77/KR7/INTP11	G5	P02/SO10/TxD1
A6	P62	C6	P60/SCL0	E6	P41/TOOL1	G6	P00/TI00
A7	P61/SDA0	C7	V_{SS}	E7	$\overline{RESET}$	G7	P140/PCLBUZ0 /INTP6
A8	EV_{DD}	C8	P121/X1	E8	FLMD0	G8	P124/XT2
B1	P51/INTP2	D1	P16/TI01/TO01 /INTP5	F1	P11/SI00/RxD0	H1	AVSS
B2	P50/INTP1	D2	P14/RxD3	F2	P12/SO00/TxD0	H2	P26/ANI6
B3	P27/ANI7	D3	P55	F3	P20/ANI0	H3	P25/ANI5
B4	P03/SI10/RxD1 /SDA10	D4	P71/KR1	F4	P130	H4	P21/ANI1
B5	P75/KR5/INTP9	D5	P72/KR2	F5	P73/KR3	H5	P04/$\overline{SCK10}$/SCL10
B6	P31/TI03/TO03/INTP4	D6	P40/TOOL0	F6	P43	H6	P01/TO00
B7	V_{DD}	D7	REGC	F7	P42/TI04/TO04	H7	P141/PCLBUZ1 /INTP7
B8	EV_{SS}	D8	P122/X2/EXCLK	F8	P123/XT1	H8	P120/INTP0/EXLVI

Note Under development

Cautions 1. Make AV_{SS} the same potential as EV_{SS} and V_{SS}.

2. Make EV_{DD} the same potential as V_{DD}.

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

Pin Identification

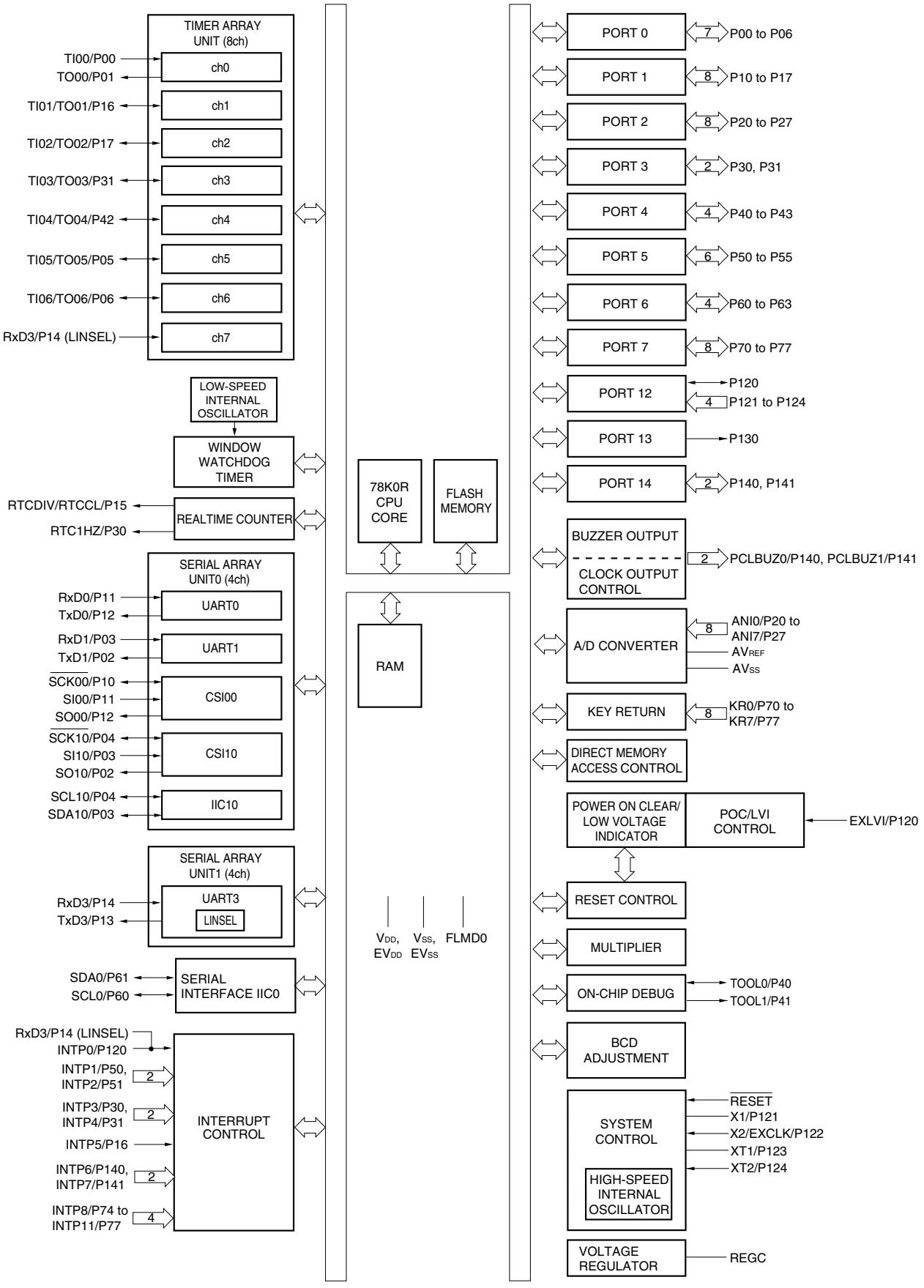
ANI0-ANI7:	Analog input
AV_{REF}:	Analog reference voltage
AV_{SS}:	Analog ground
EV_{DD}:	Power supply for port
EV_{SS}:	Ground for port
EXCLK:	External clock input (main system clock)
EXLVI:	External potential input for low-voltage detector
FLMD0:	Flash programming mode
INTP0-INTP11:	External interrupt input
KR0-KR7:	Key return
P00-P06:	Port 0
P10-P17:	Port 1
P20-P27:	Port 2
P30, P31:	Port 3
P40-P43:	Port 4
P50-P55:	Port 5
P60-P63:	Port 6
P70-P77:	Port 7
P120-P124:	Port 12
P130:	Port 13
P140, P141:	Port 14
PCLBUZ0, PCLBUZ1:	Programmable clock output/ buzzer output
REGC:	Regulator capacitance
$\overline{\text{RESET}}$:	Reset
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, RxD1, RxD3:	Receive data
$\overline{\text{SCK00}}$, $\overline{\text{SCK10}}$:	Serial clock input/output
SCL0, SCL10:	Serial clock input/output
SDA0, SDA10:	Serial data input/output
SI00, SI10:	Serial data input
SO00, SO10:	Serial data output
TI00-TI06:	Timer input
TO00-TO06:	Timer output
TOOL0:	Data input/output for tool
TOOL1:	Clock output for tool
TxD0, TxD1, TxD3:	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 Microcontroller Lineup

ROM	RAM	78K0R/KE3	78K0R/KF3	78K0R/KG3	78K0R/KH3	78K0R/KJ3
		64 Pins	80 Pins	100 Pins	128 Pins	144 Pins
512 KB	30 KB	–	–	μPD78F1168[Note]	μPD78F1178[Note]	μPD78F1188[Note]
384 KB	24 KB	–	–	μPD78F1167[Note]	μPD78F1177[Note]	μPD78F1187[Note]
256 KB	12 KB	μPD78F1146	μPD78F1156	μPD78F1166	μPD78F1176[Note]	μPD78F1186[Note]
192 KB	10 KB	μPD78F1145	μPD78F1155	μPD78F1165	μPD78F1175[Note]	μPD78F1185[Note]
128 KB	8 KB	μPD78F1144	μPD78F1154	μPD78F1164	μPD78F1174[Note]	μPD78F1184[Note]
96 KB	6 KB	μPD78F1143	μPD78F1153	μPD78F1163	–	–
64 KB	4 KB	μPD78F1142	μPD78F1152	μPD78F1162	–	–

Note Under development

1.6 Block Diagram



1.7 Outline of Functions

(1/2)

Item		μPD78F1142	μPD78F1143	μPD78F1144	μPD78F1145	μPD78F1146
Internal memory	Flash memory (self-programming supported)	64 KB	96 KB	128 KB	192 KB	256 KB
	RAM	4 KB	6 KB	8 KB	10 KB	12 KB
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, 2 to 5 MHz: V_{DD} = 1.8 to 5.5 V				
	Internal high-speed oscillation clock	Internal oscillation 8 MHz (TYP.): V_{DD} = 1.8 to 5.5 V				
Subsystem clock (Oscillation frequency)		XT1 (crystal) oscillation 32.768 kHz (TYP.): V_{DD} = 1.8 to 5.5 V				
Internal low-speed oscillation clock (For WDT)		Internal oscillation 240 kHz (TYP.): V_{DD} = 1.8 to 5.5 V				
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)				
		0.125 μs (Internal high-speed oscillation clock: f_{IH} = 8 MHz (TYP.) operation)				
		61 μs (Subsystem clock: f_{SUB} = 32.768 kHz operation)				
Instruction set		• 8-bit operation, 16-bit operation • Multiply (16 bits × 16 bits) • Bit manipulation (Set, reset, test, and Boolean operation), etc.				
I/O port		Total: 55 CMOS I/O: 46 CMOS input: 4 CMOS output: 1 N-ch open-drain I/O (6 V tolerance): 4				
Timer		• 16-bit timer: 8 channels • Watchdog timer: 1 channel • Real-time counter: 1 channel				
	Timer outputs	7 (PWM output: 6)				
	RTC outputs	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)				
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)				
A/D converter		10-bit resolution × 8 channels (AV_{REF} = 2.3 to 5.5 V)				

(2/2)

Item		μPD78F1142	μPD78F1143	μPD78F1144	μPD78F1145	μPD78F1146
Serial interface		• UART supporting LIN-bus: 1 channel • UART/CSI: 1 channel • UART/CSI/simplified I^2C: 1 channel • I^2C bus: 1 channel				
Multiplier		16 bits × 16 bits = 32 bits				
DMA controller		2 channels				
Vectored interrupt sources	Internal	25				
	External	13				
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 1]				
On-chip debug function		Provided				
Power supply voltage		V_{DD} = 1.8 to 5.5 V				
Operating ambient temperature		T_A = –40 to +85°C				
Package		64-pin plastic LQFP (12 × 12) (0.65 mm pitch) 64-pin plastic LQFP (fine pitch) (10 × 10) (0.5 mm pitch) <R> 64-pin plastic TQFP (fine pitch) (7 × 7) (0.4 mm pitch)[Note 2] <R> 64-pin plastic FBGA (5 × 5) (0.5 mm pitch)[Note 2]				

Notes 1. 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.

2. Under development

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

There are three types of pin I/O buffer power supplies: AV_{REF}, EV_{DD}, and V_{DD}. The relationship between these power supplies and the pins is shown below.

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

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

(1) Port functions (1/2)

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				TI05/TO05
P06				TI06/TO06
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{\text{SCK00}}$
P11				SI00/RxD0
P12				SO00/TxD0
P13				TxD3
P14				RxD3
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. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOOL0
P41				TOOL1
P42				TI04/TO04
P43				–
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				–
P53				–
P54				–
P55				–
P60	I/O	Port 6. 4-bit I/O port. Output of P60 to P63 can be set to N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCL0
P61				SDA0
P62				–
P63				–
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 to P77				KR4/INTP8 to KR7/INTP11

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 P43 (port 4)**).

(1) Port functions (2/2)

Function Name	I/O	Function	After Reset	Alternate Function
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. 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	PCLBUZ0/INTP6
P141				PCLBUZ1/INTP7

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

Function Name	I/O	Function	After Reset	Alternate Function
ANI0-ANI7	Input	A/D converter analog input	Digital input port	P20 to P27
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
INTP6				P140/PCLBUZ0
INTP7				P141/PCLBUZ1
INTP8				P74/KR4 to P77/KR7
INTP9				
INTP10				
INTP11				
KR0-KR3	Input	Key interrupt input	Input port	P70 to P73
KR4-KR7				P74/INTP8 to P77/INTP11
PCLBUZ0	Output	Clock output/buzzer output	Input port	P140/INTP6
PCLBUZ1				P141/INTP7
REGC	–	Connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF).	–	–
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
RxD3	Input	Serial data input to UART3	Input port	P14
$\overline{\text{SCK00}}$	I/O	Clock input/output for CSI00 and CSI10	Input port	P10
$\overline{\text{SCK10}}$				P04/SCL10
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/$\overline{\text{SCK10}}$
SDA0	I/O	Serial data I/O for I^2C	Input port	P61
SDA10		Clock input/output for simplified I^2C	Input port	P03/SI10/RxD1
SI00	I/O	Serial data input to CSI00 and CSI10	Input port	P11/RxD0
SI10				P03/RxD1/SDA10
SO00	Output	Serial data output from CSI00 and CSI10	Input port	P12/TxD0
SO10				P02/TxD1

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

Function Name	I/O	Function	After Reset	Alternate Function
TI00	Input	External count clock input to 16-bit timer 00	Input port	P00
TI01		External count clock input to 16-bit timer 01		P16/TO01/INTP5
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
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		P05/TI05
TO06		16-bit timer 06 output		P06/TI06
TxD0	Output	Serial data output from UART0	Input port	P12/SO00
TxD1	Output	Serial data output from UART1	Input port	P02/SO10
TxD3	Output	Serial data output from UART3	Input port	P13
X1	–	Resonator connection for main system clock	Input port	P121
X2	–		Input port	P122/EXCLK
EXCLK	Input	External clock input for main system clock	Input port	P122/X2
XT1	–	Resonator connection for subsystem clock	Input port	P123
XT2	–		Input port	P124
V_{DD}	–	Positive power supply (P121 to P124 and other than ports)	–	–
EV_{DD}	–	Positive power supply for ports (other than P20 to P27, P121 to P124)	–	–
AV_{REF}	–	• A/D converter reference voltage input • Positive power supply for P20 to P27, and A/D converter	–	–
V_{SS}	–	Ground potential (P121 to P124 and other than ports)	–	–
EV_{SS}	–	Ground potential for ports (other than P20 to P27 and P121 to P124)	–	–
AV_{SS}	–	Ground potential for A/D converter, P20 to P27. 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

2.2 Description of Pin Functions

2.2.1 P00 to P06 (port 0)

P00 to P06 function as a 7-bit 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).

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

(1) Port mode

P00 to P06 function as a 7-bit 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

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

2.2.2 P10 to P17 (port 1)

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

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

(1) Port mode

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

(2) Control mode

P10 to P17 function as external interrupt request input, serial interface data I/O, clock I/O, timer I/O, 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) RxD3

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

(f) TxD0

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

(g) TxD3

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

(h) TI01, TI02

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

(i) TO01, TO02

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

(j) INTP5

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

(k) RTCDIV

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

(l) RTCCL

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

Cautions 1. To use P10/$\overline{\text{SCK00}}$ and P12/SO00/TxD0 as general-purpose ports, set serial communication operation setting register 00 (SCR00) to the default status (0087H).

2. Do not enable outputting RTCCL and RTCDIV at the same time.

2.2.3 P20 to P27 (port 2)

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

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

(1) Port mode

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

(2) Control mode

P20 to P27 function as A/D converter analog input pins (ANI0 to ANI7). When using these pins as analog input pins, see **10.6 (5) ANI0/P20 to ANI7/P27**.

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 P43 (port 4)

P40 to P43 function as an 4-bit I/O port. These pins also function as data I/O for a flash memory programmer/debugger, clock output, and timer I/O.

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

(1) Port mode

P40 to P43 function as an 4-bit I/O port. P40 to P43 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 P43 function as data I/O for a flash memory programmer/debugger, clock output, and timer I/O.

(a) 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).

(b) 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).

(c) TI04

This is a pin for inputting an external count clock/capture trigger to 16-bit timers 04.

(d) TO04

This is a timer output pin from 16-bit timers 04.

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

(a) In normal operation mode and when on-chip debugging is disabled (OCDENSET = 0) by an option byte (000C3H)
=> Use this pin as a port pin (P40).

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

=> Connect this pin to EV_{DD} 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_{DD} via an external resistor.

2.2.6 P50 to P55 (port 5)

P50 to P55 function as an 6-bit I/O port. These pins also function as external interrupt request input.

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

(1) Port mode

P50 to P55 function as an 6-bit I/O port. P50 to P55 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 P55 function as external interrupt request 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.

2.2.7 P60 to P63 (port 6)

P60 to P63 function as an 4-bit I/O port. These pins also function as serial interface data I/O and clock I/O.

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

(1) Port mode

P60 to P63 function as an 4-bit I/O port. P60 to P63 can be set to input port or output port in 1-bit units using port mode register 6 (PM6).

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

(2) Control mode

P60 to P63 function as serial interface data I/O and clock I/O.

(a) SDA0

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

(b) SCL0

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

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 to INTP11

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

2.2.9 P120 to P124 (port 12)

P120 functions as a 1-bit I/O port. P121 to P124 function as a 4-bit input port. These pins also function as external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, 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 function 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.

2.2.10 P130 (port 13)

P130 functions as a 1-bit output port.

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 **4.2.10 Port 13**).

2.2.11 P140, P141 (port 14)

P140 and P141 function as a 2-bit I/O port. These pins also function as external interrupt request input and clock/buzzer output.

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

(1) Port mode

P140 and P141 function as a 2-bit I/O port. P140 and P141 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 and P141 function as external interrupt request input, and clock/buzzer output.

(a) INTP6, 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) PCLBUZ0, PCLBUZ1

These are the clock/buzzer output pins.

2.2.12 AV_{REF}

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

When all pins of port 2 are used as the analog port pins, make the potential of AV_{REF} be such that 2.3 V ≤ AV_{REF} ≤ V_{DD}. When one or more of the pins of port 2 are used as the digital port pins or when the A/D converter is not used, make AV_{REF} the same potential as EV_{DD} or V_{DD}.

2.2.13 AV_{SS}

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

2.2.14 $\overline{\text{RESET}}$

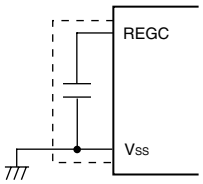
This is the active-low system reset input pin.

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

2.2.15 REGC

This is the pin for connecting regulator output (2.5 V) stabilization capacitance for internal operation. Connect this
<R> 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.



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

2.2.16 V_{DD}, EV_{DD}

V_{DD} is the positive power supply pin for P121 to P124 and other than ports.

EV_{DD} is the positive power supply pin for ports other than P20 to P27 and P121 to P124.

2.2.17 V_{SS}, EV_{SS}

V_{SS} is the ground potential pin for P121 to P124 and other than ports.

EV_{SS} is the ground potential pin for ports other than P20 to P27 and P121 to P124.

<R> **2.2.18 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 **23.5 (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

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

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

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/TI00	8-R	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01/TO00	5-AG		
P02/SO10/TxD1			
P03/SI10/RxD1/SDA10	5-AN		
P04/$\overline{SCK10}$/SCL10			
P05/TI05/TO05	8-R		
P06/TI05/TO05			
P10/$\overline{SCK00}$			
P11/SI00/RxD0			
P12/SO00/TxD0	5-AG		
P13/TxD3			
P14/RxD3	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_{DD} or EV_{SS} via a resistor. Output: Leave open.
P42/TI04/TO04	8-R		
P43	5-AN		
P50/INTP1, P51/INTP2	8-R		
P52 to P55	5-AG		
P60/SCL0	13-R		Input: Independently connect to EV_{SS}. Output: Set the port output latch to 0 and leave these pins open via low-level output.
P61/SDA0			
P62, P63	13-P		
P70/KR0 to P73/KR3	8-R		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P74/KR4/INTP8 to P77/KR7/INTP11			
P120/INTP0/EXLVI			

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

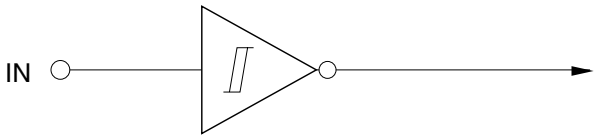
Table 2-2. Connection of Unused Pins (2/2)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
<R> P121/X1[Note]	37-B	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P122/X2/EXCLK[Note]			
P123/XT1[Note]			
P124/XT2[Note]			
P130	3-C	Output	Leave open.
P140/PCLBUZ0/INTP6	8-R	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P141/PCLBUZ1/INTP7			
AV_{REF}	–	–	<When one or more of P20 to P27 are set as a digital port> Make this pin the same potential as EV_{DD} or V_{DD}. <When all of P20 to P27 are set as analog ports> Make this pin to have a potential where 2.3 V ≤ AV_{REF} ≤ V_{DD}.
AV_{SS}	–	–	Make this pin the same potential as the EV_{SS} or V_{SS}.
<R> 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 to V_{DD} or via a resistor.
REGC	–	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).

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.

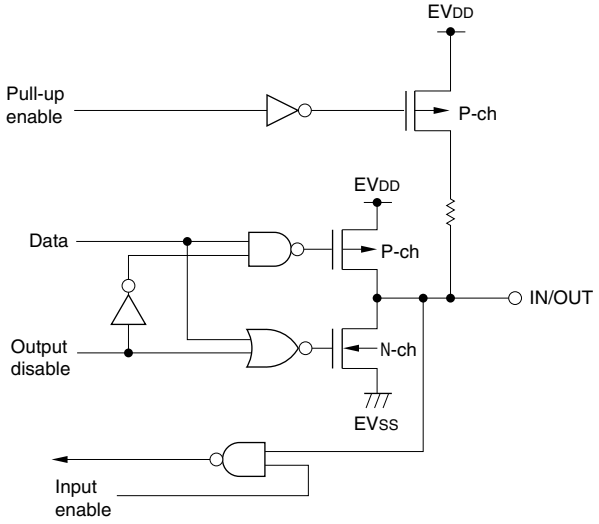
Figure 2-1. Pin I/O Circuit List (1/2)

Type 2



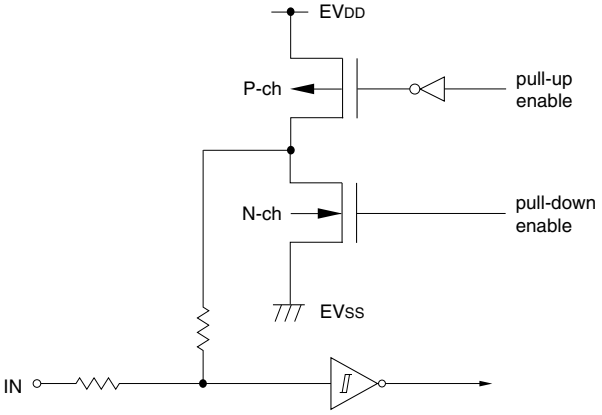
Schmitt-triggered input with hysteresis characteristics

Type 5-AG



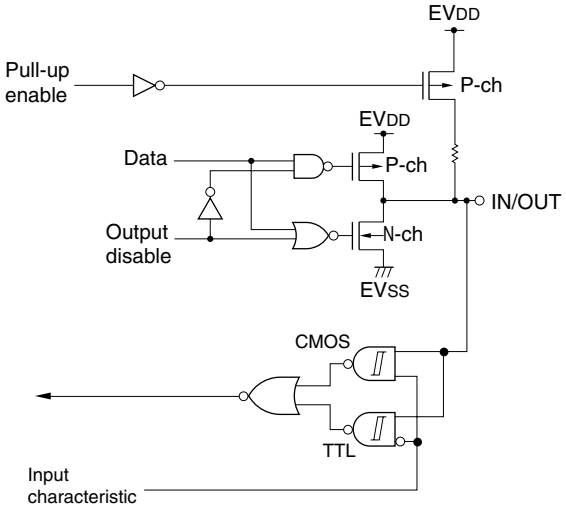
<R>

Type 2-W

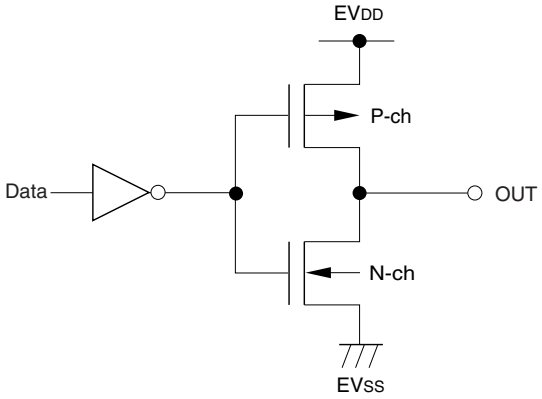


Schmitt-triggered input with hysteresis characteristics

Type 5-AN



Type 3-C



Type 8-R

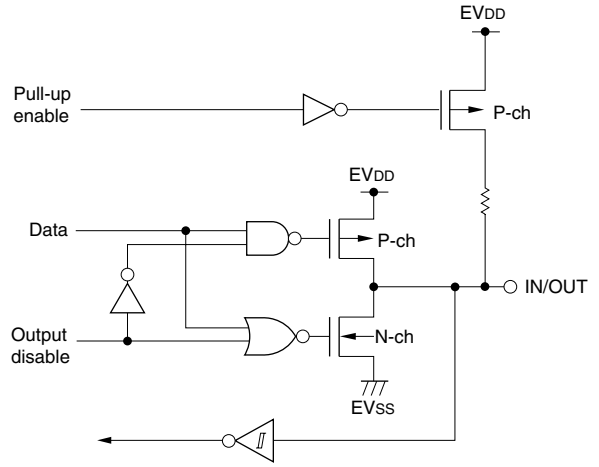
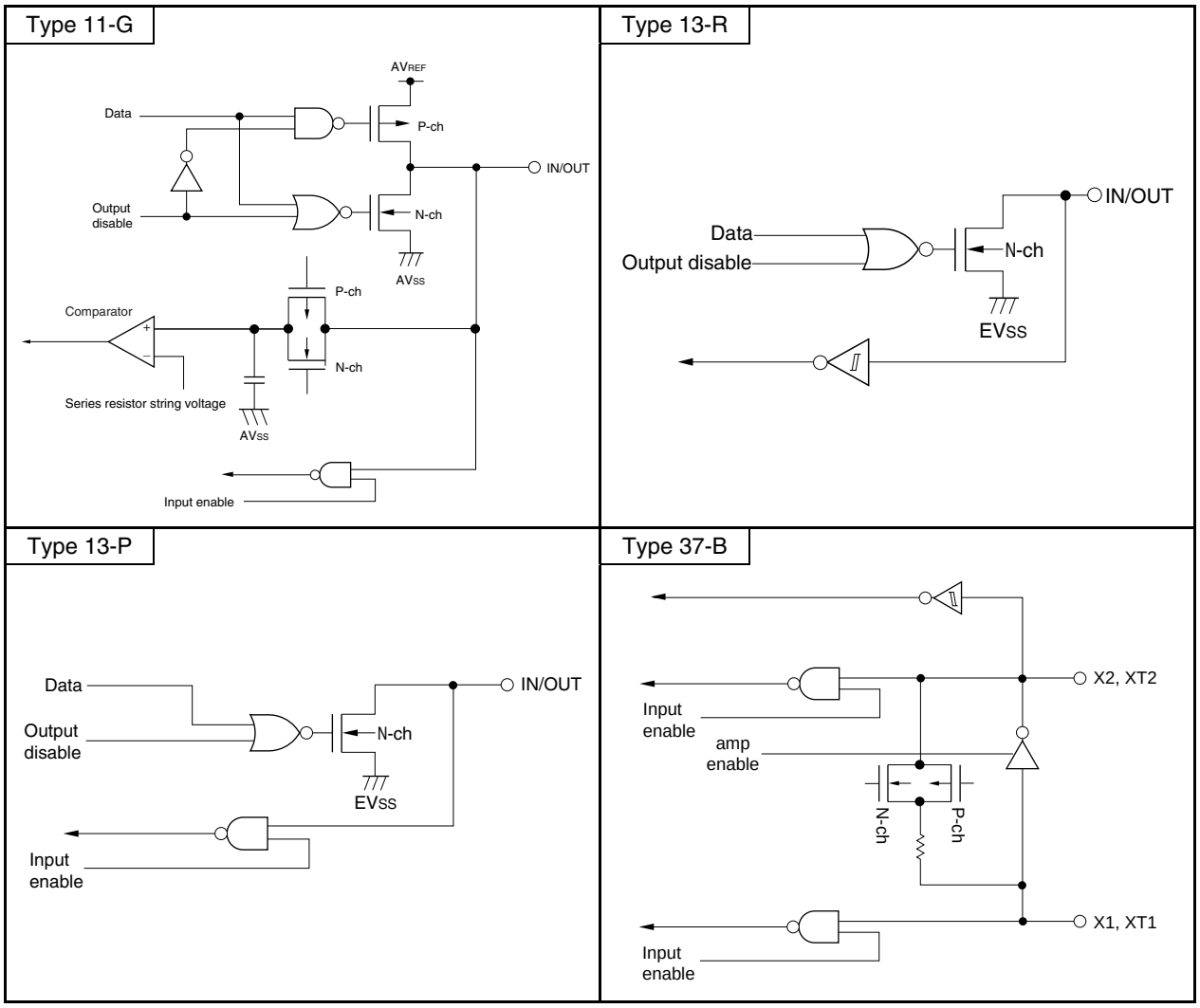


Figure 2-1. Pin I/O Circuit List (2/2)

<R>

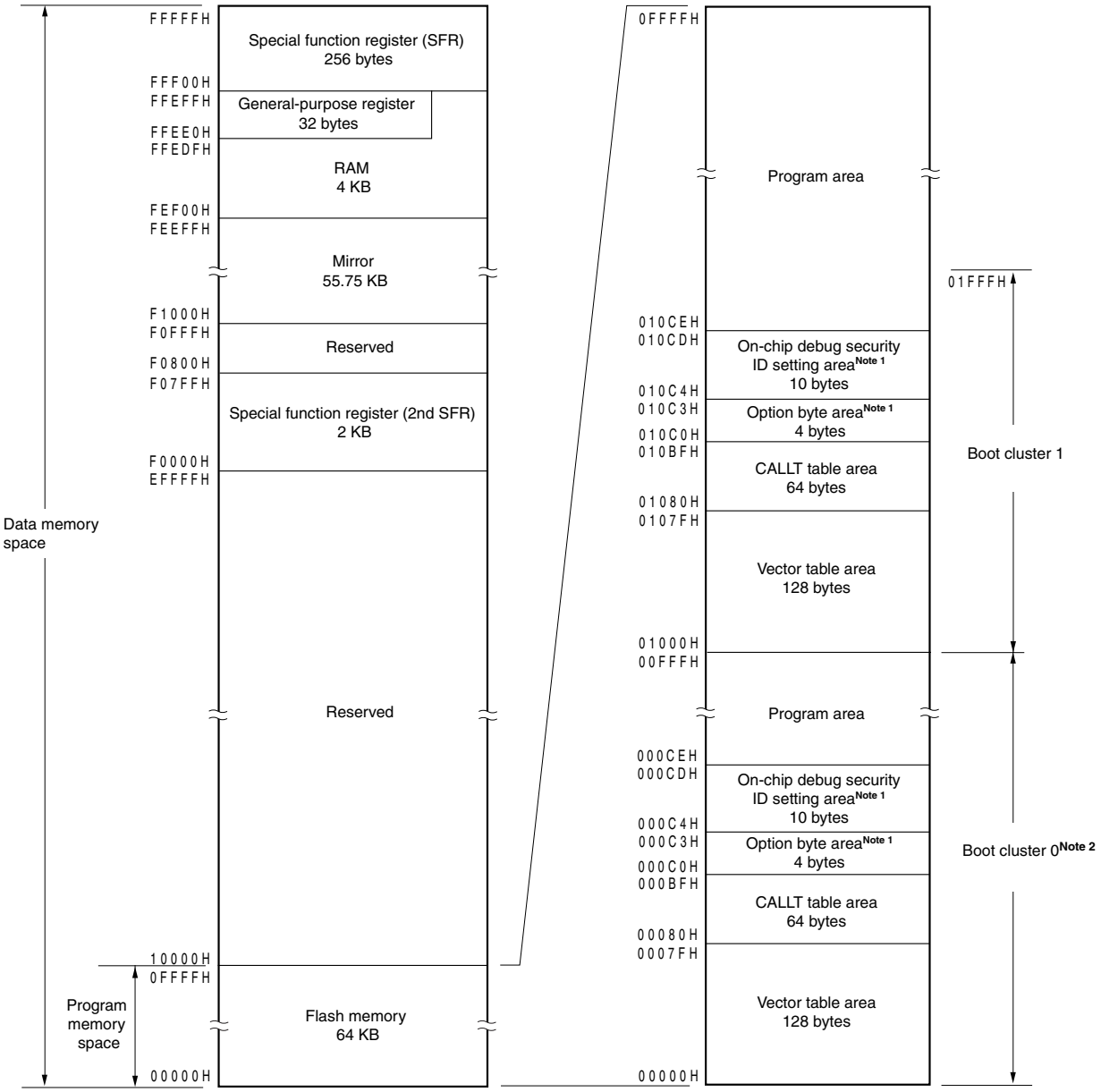


CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the 78K0R/KE3 can access a 1 MB memory space. Figures 3-1 to 3-5 show the memory maps.

Figure 3-1. Memory Map (μPD78F1142)

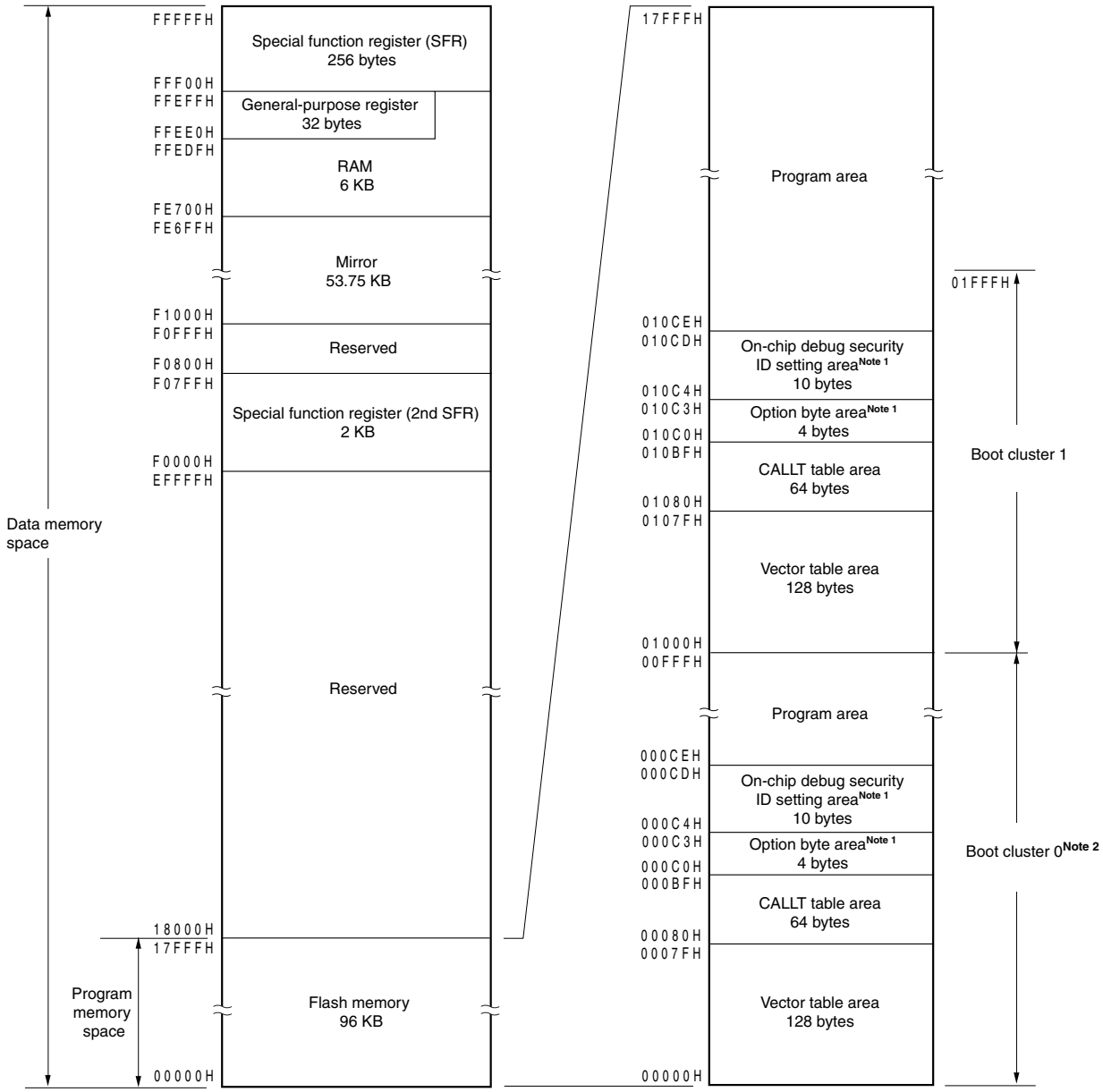


Notes 1. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **23.7 Security Setting**).

Figure 3-2. Memory Map (μPD78F1143)

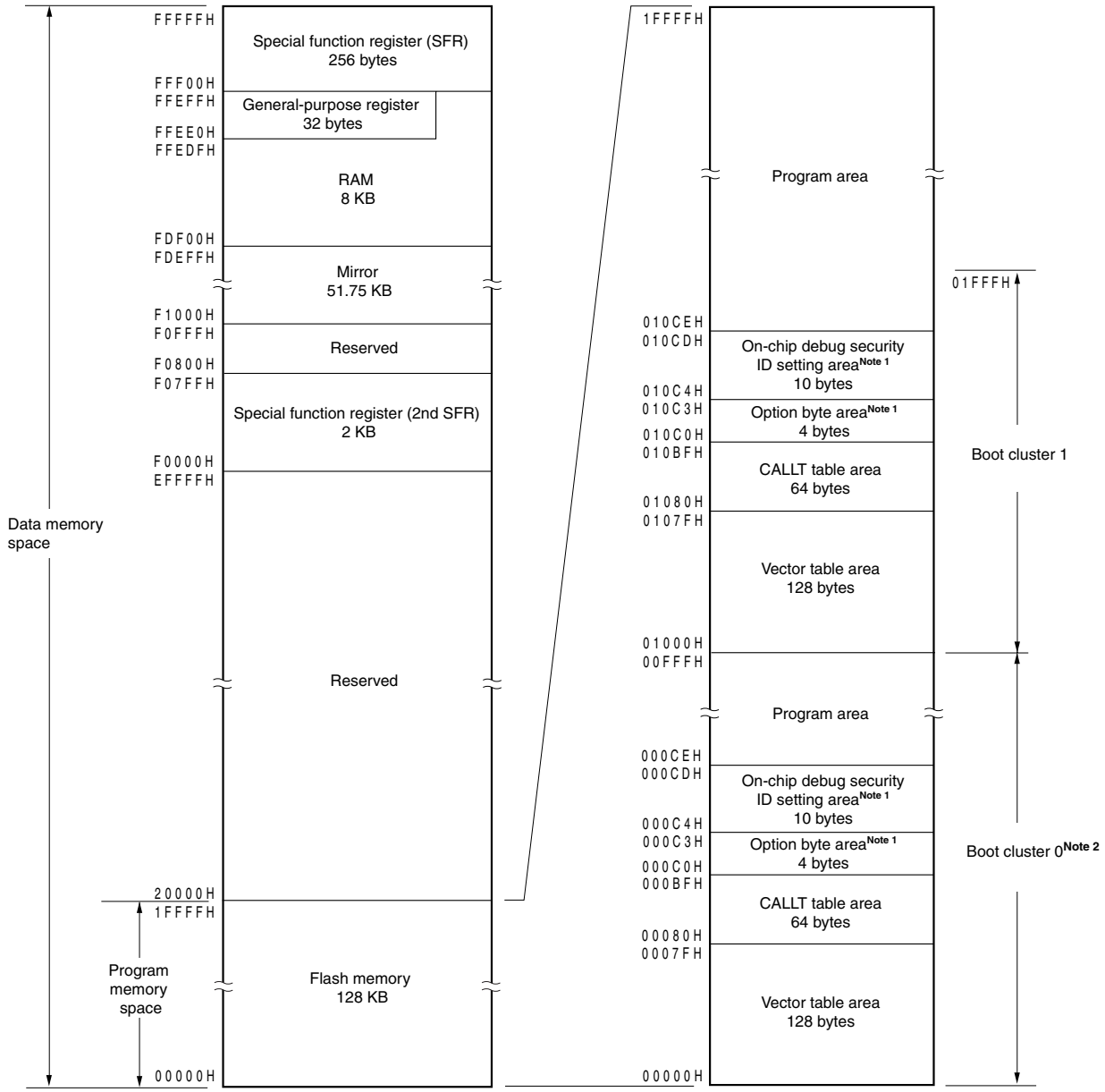


Notes 1. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **23.7 Security Setting**).

Figure 3-3. Memory Map (μPD78F1144)

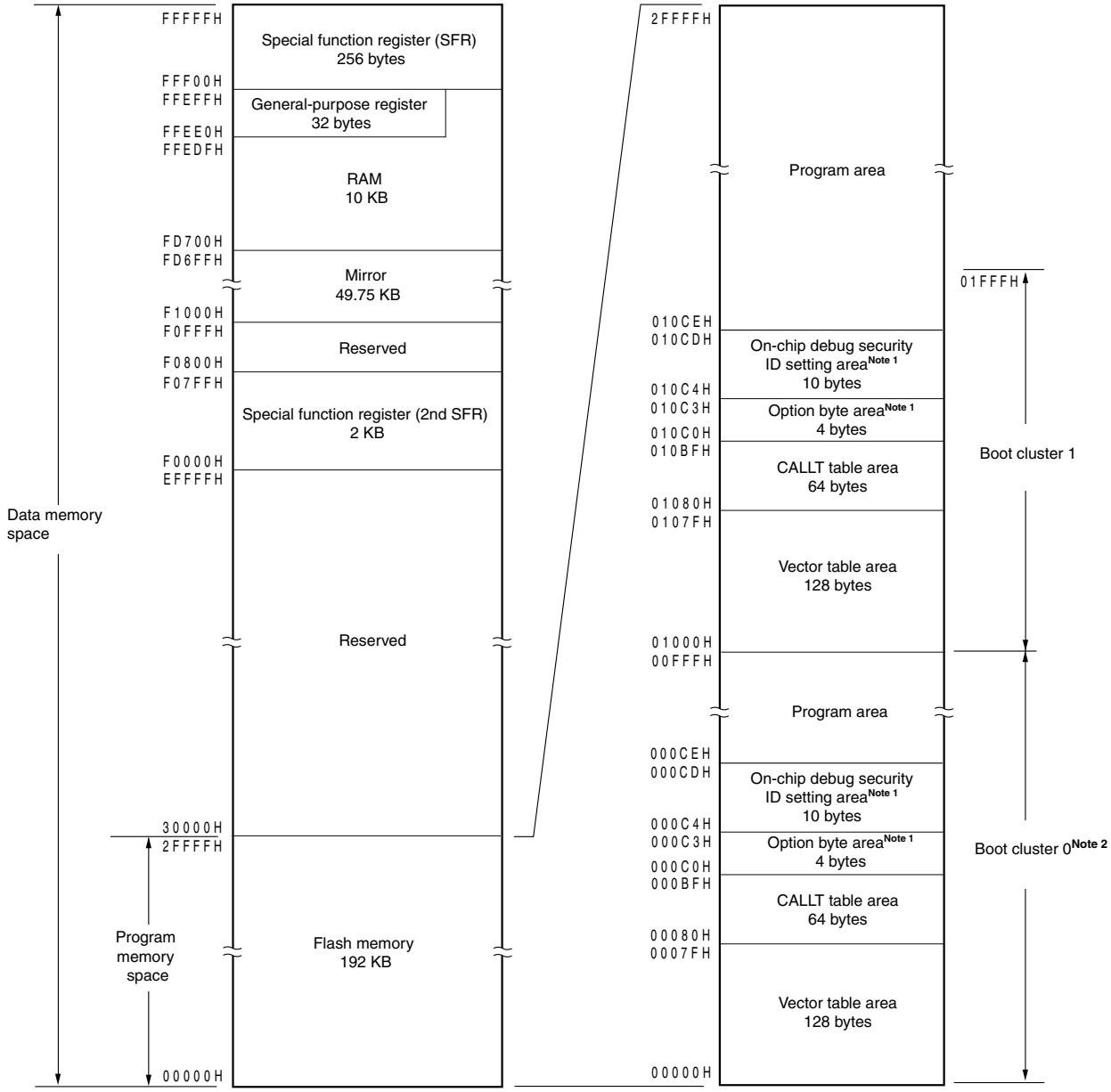


Notes 1. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **23.7 Security Setting**).

Figure 3-4. Memory Map (μPD78F1145)

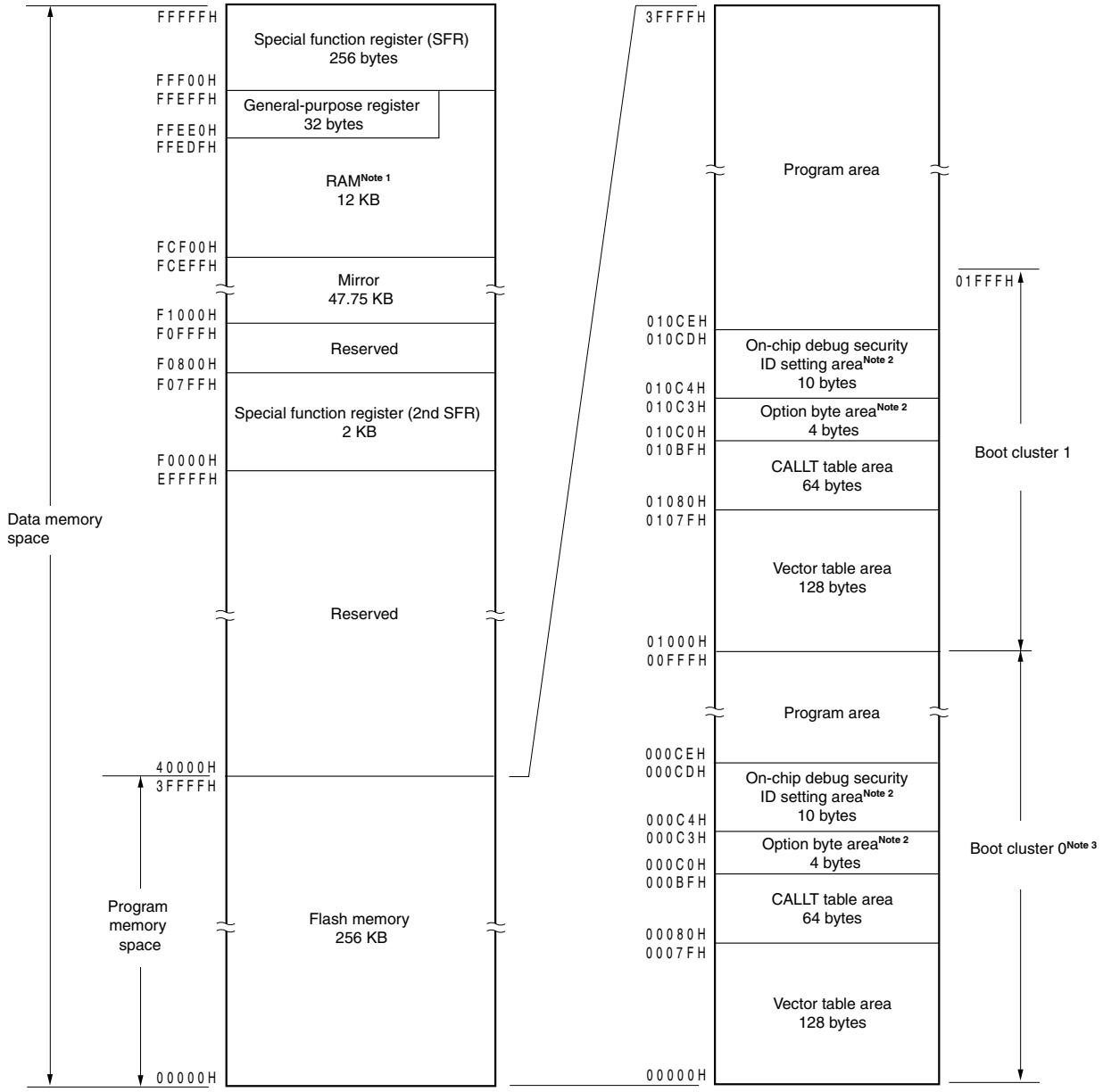


Notes 1. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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.

2. Writing boot cluster 0 can be prohibited depending on the setting of security (see **23.7 Security Setting**).

Figure 3-5. Memory Map (μPD78F1146)



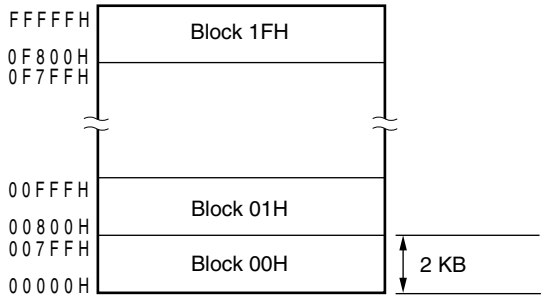
Notes 1. Use of the area FCF00H to FD6FFH is prohibited when using the self-programming function. Since this area is used for self-programming library.

2. 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.

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **23.7 Security Setting**).

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



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

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

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

Remark μPD78F1142: Block numbers 00H to 1FH
μPD78F1143: Block numbers 00H to 2FH
μPD78F1144: Block numbers 00H to 3FH
μPD78F1145: Block numbers 00H to 5FH
μPD78F1146: Block numbers 00H to 7FH

3.1.1 Internal program memory space

The internal program memory space stores the program and table data. Normally, it is addressed with the program counter (PC).

78K0R/KE3 products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F1142	Flash memory	65536 × 8 bits (00000H to 0FFFFH)
μPD78F1143		98303 × 8 bits (00000H to 17FFFH)
μPD78F1144		131071 × 8 bits (00000H to 1FFFFH)
μPD78F1145		196607 × 8 bits (00000H to 2FFFFH)
μPD78F1146		262143 × 8 bits (00000H to 3FFFFH)

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

(1) Vector table area

The 128-byte area 00000H to 0007FH is reserved as a vector table area. The program start addresses for branch upon reset or generation of each interrupt request are stored in the vector table area.

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	0002AH	INTIIC0
		0002CH	INTTM00
00004H	INTWDTI	0002EH	INTTM01
00006H	INTLVI	00030H	INTTM02
00008H	INTP0	00032H	INTTM03
0000AH	INTP1	00034H	INTAD
0000CH	INTP2	00036H	INTRTC
0000EH	INTP3	00038H	INTRTCI
00010H	INTP4	0003AH	INTKR
00012H	INTP5	00042H	INTTM04
00014H	INTST3	00044H	INTTM05
00016H	INTSR3	00046H	INTTM06
00018H	INTSRE3	00048H	INTTM07
0001AH	INTDMA0	0004AH	INTP6
0001CH	INTDMA1	0004CH	INTP7
0001EH	INTST0/INTCSI00	0004EH	INTP8
00020H	INTSR0	00050H	INTP9
00022H	INTSRE0	00052H	INTP10
00024H	INTST1/INTCSI10/INTIIC10	00054H	INTP11
00026H	INTSR1	0007EH	BRK
00028H	INTSRE1		

(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 22 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 24 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The μPD78F1142 mirrors the data flash area of 00000H to 0FFFFH, to F0000H to FFFFFH.

The μPD78F1143, 78F1144, 78F1145, and 78F1146 mirrors the data flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, to F0000H to FFFFFH (the data flash area to be mirrored is set by the processor mode control register (PMC)).

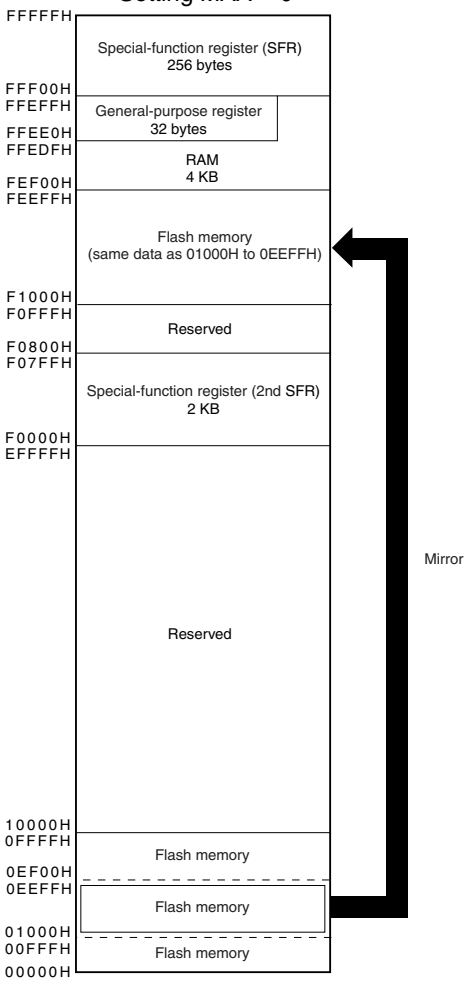
By reading data from F0000H to FFFFFH, the contents of the data flash can be read with the shorter code. However, the data flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

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

The following show examples.

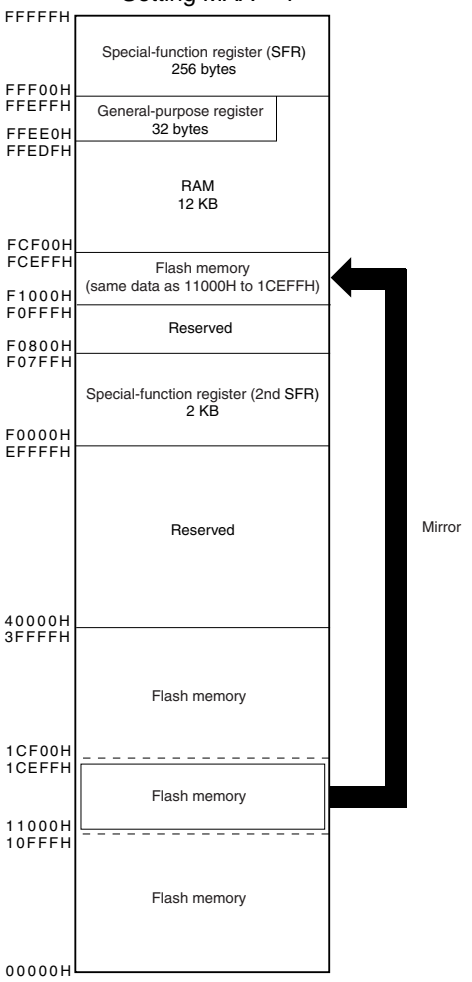
Example 1 μPD78F1142
(Flash memory: 64 KB, RAM: 4 KB)

Setting MAA = 0



Example 2 μPD78F1146
(Flash memory: 256 KB, RAM: 12 KB)

Setting MAA = 1



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

PMC register is described below.

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

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

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

Reset signal generation sets this register to 00H.

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

Address: FFFFEH After reset: 00H R/W

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

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

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

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

3. When the μPD78F1142 is used, be sure to set bit 0 (MAA) of this register to 0.

3.1.3 Internal data memory space

78K0R/KE3 products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
μPD78F1142	4096 × 8 bits (FEF00H to FFEFFH)
μPD78F1143	6144 × 8 bits (FE700H to FFEFFH)
μPD78F1144	8192 × 8 bits (FDF00H to FFEFFH)
μPD78F1145	10240 × 8 bits (FD700H to FFEFFH)
μPD78F1146	12288 × 8 bits (FCF00H to FFEFFH)

The 32-byte area FFEE0H to FFEFFH is assigned to four general-purpose register banks consisting of eight 8-bit registers per bank.

This area can be used as a program area where instructions are written and executed. However, executing instructions is disabled in the general-purpose register.

The internal high-speed RAM can also be used as a stack memory.

Caution While using the self-programming function, the areas FFE20H to FFEDFH and FCF00H to FD6FFH (μPD78F1146) cannot be used as stack memory.

3.1.4 Special function register (SFR) area

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

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

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

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

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

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

3.1.6 Data memory addressing

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

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the 78K0R/KE3, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Figures 3-7 to 3-11 show correspondence between data memory and addressing.

Figure 3-7. Correspondence Between Data Memory and Addressing (μPD78F1142)

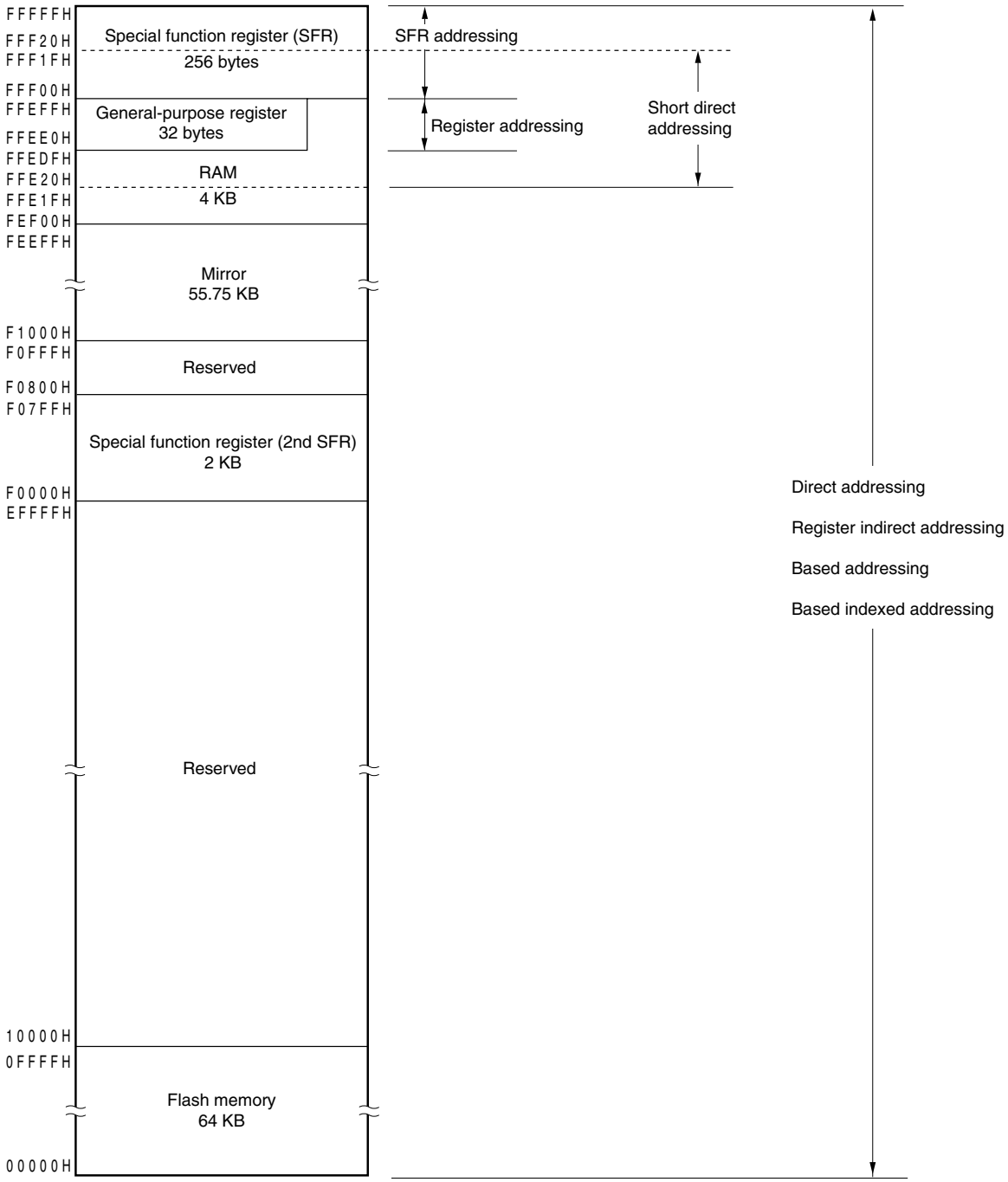


Figure 3-8. Correspondence Between Data Memory and Addressing (μPD78F1143)

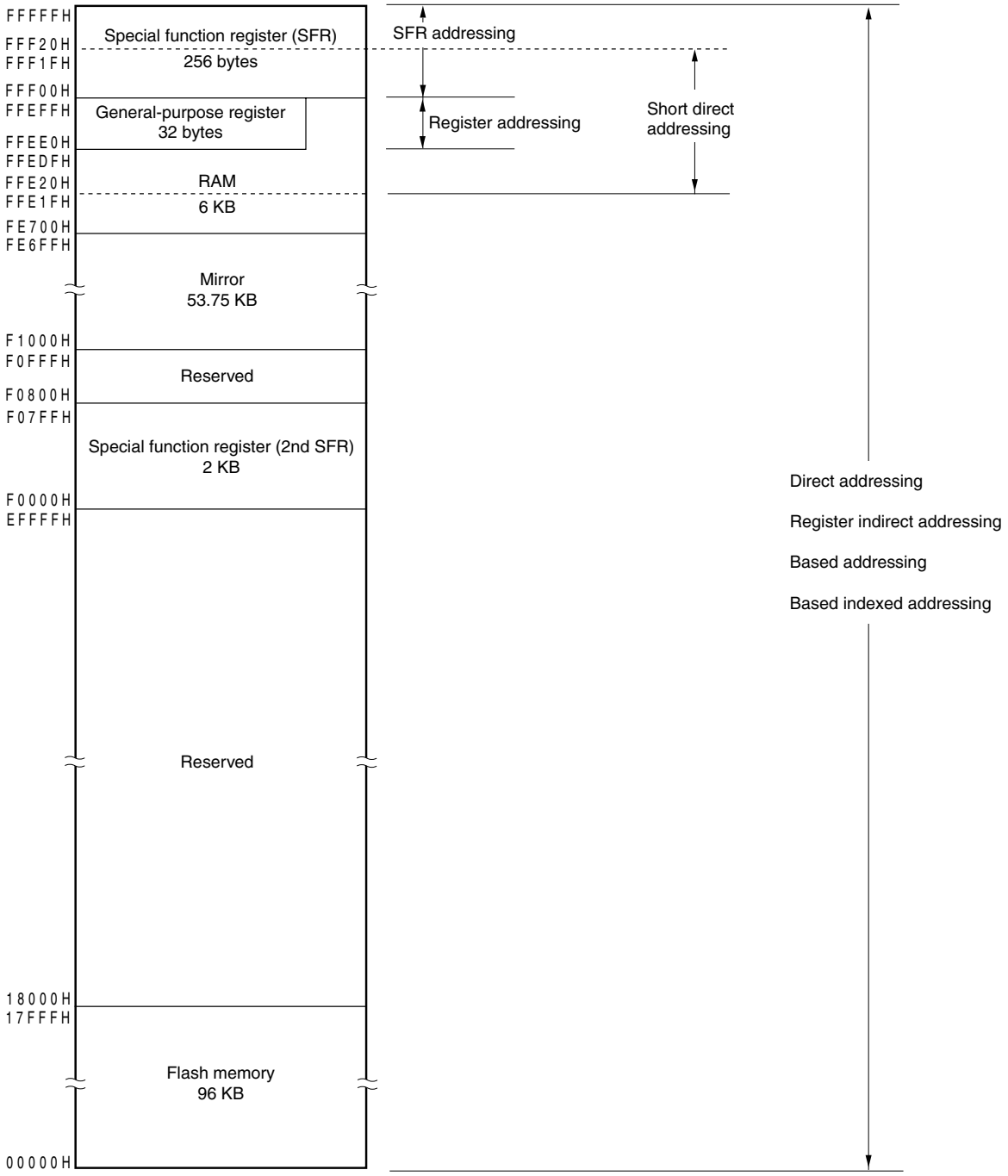


Figure 3-9. Correspondence Between Data Memory and Addressing (μPD78F1144)

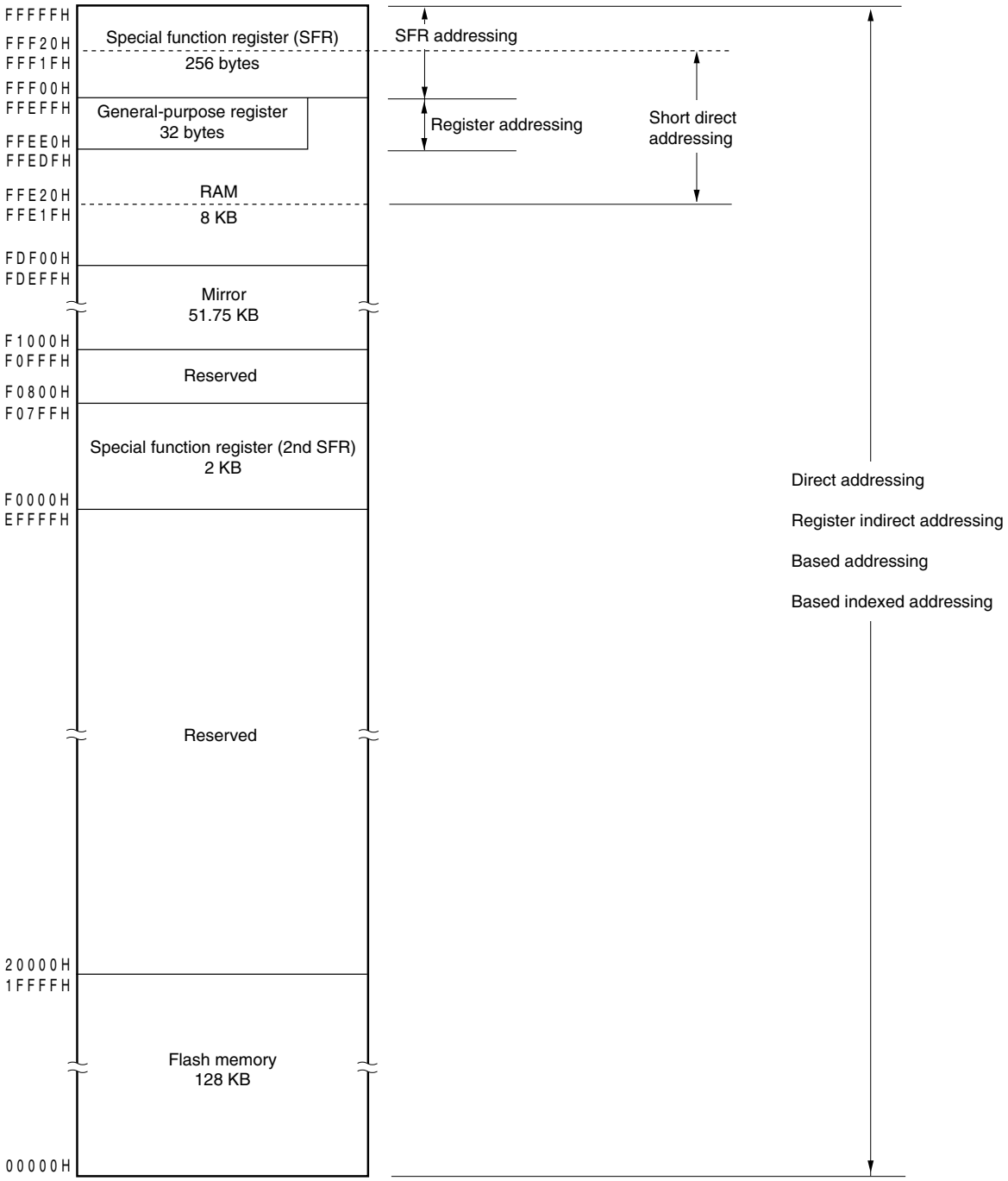


Figure 3-10. Correspondence Between Data Memory and Addressing (μPD78F1145)

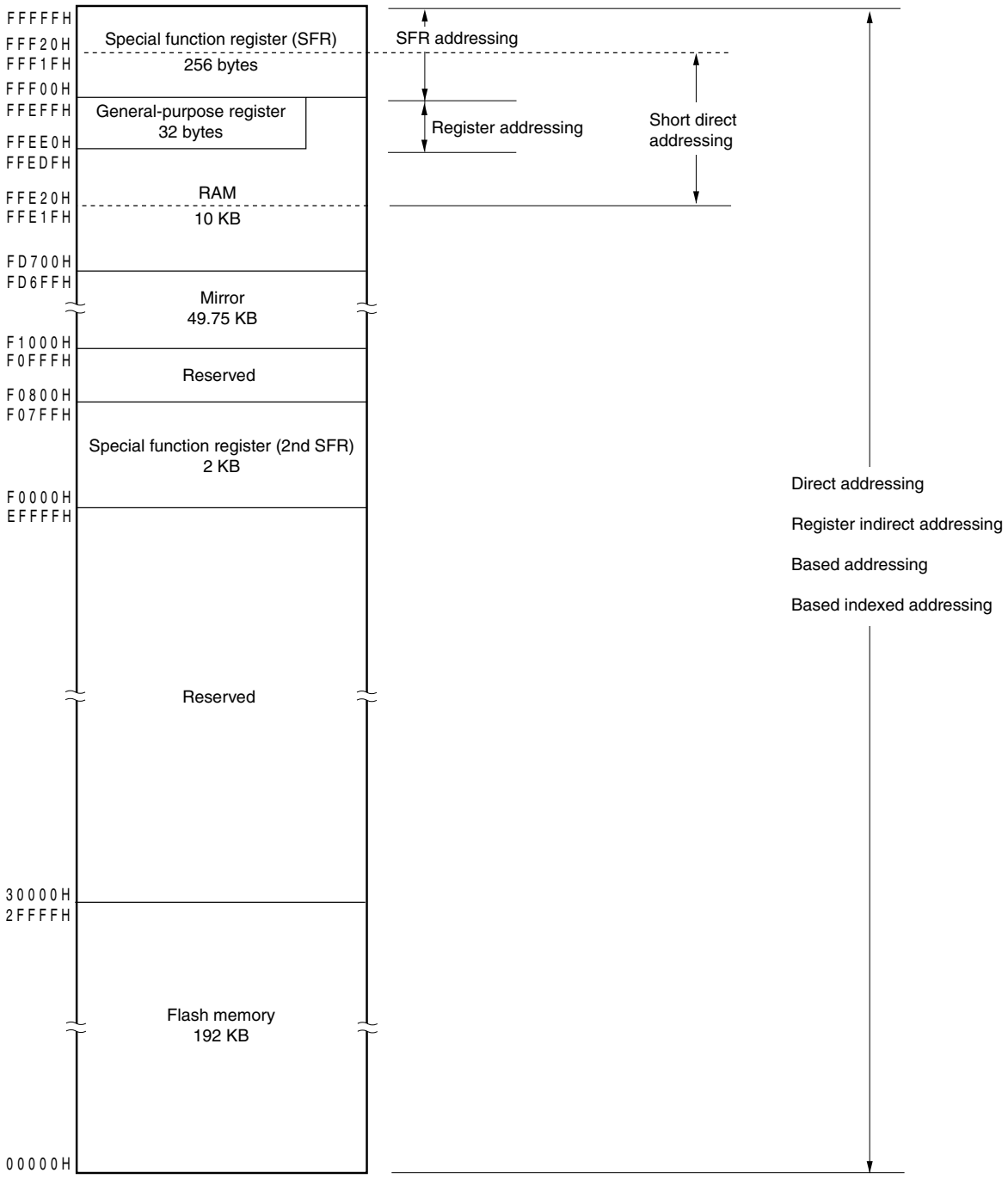
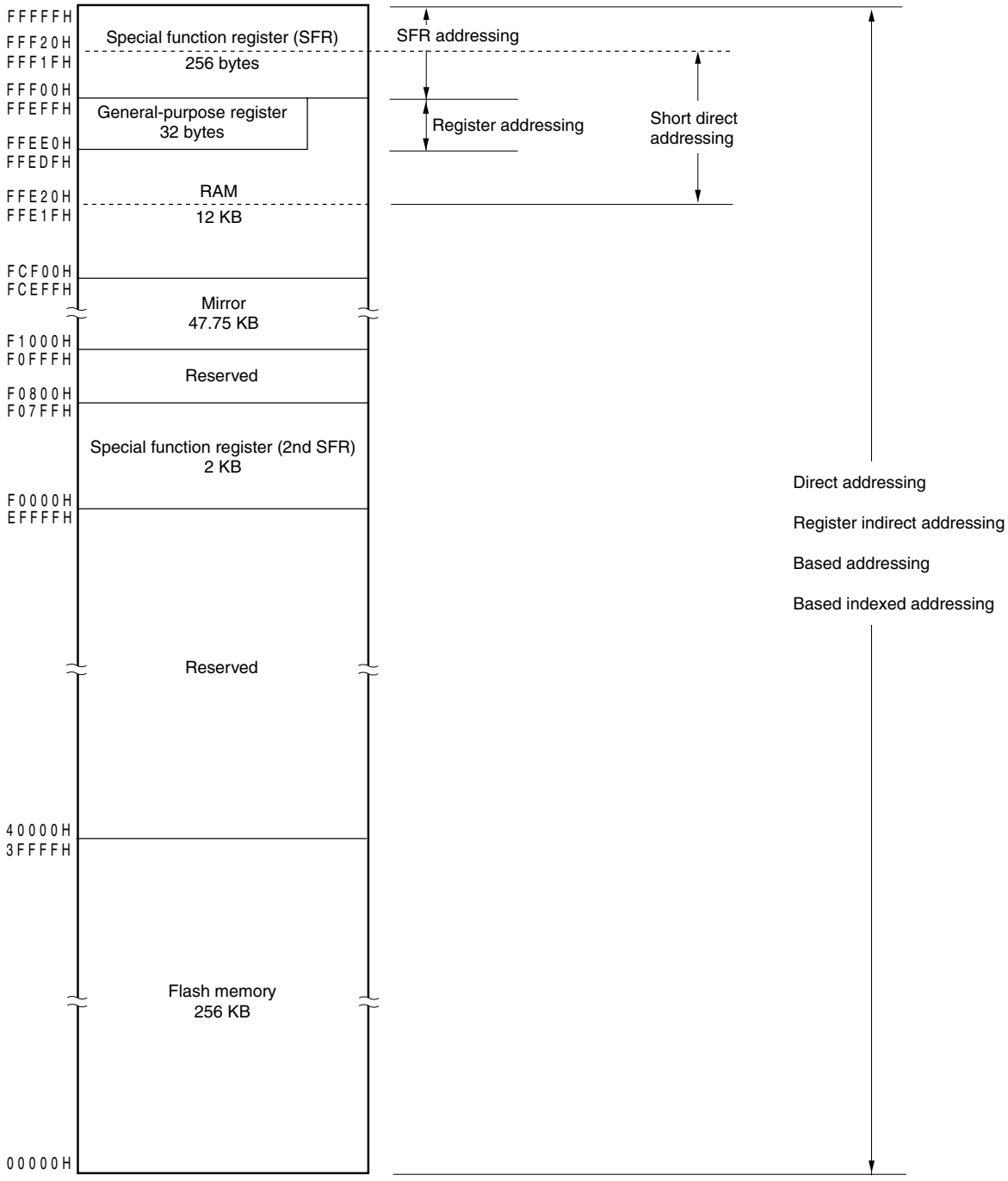


Figure 3-11. Correspondence Between Data Memory and Addressing (μPD78F1146)



Note Use of the area FCF00H to FD6FFH is prohibited when using the self-programming function. Since this area is used for self-programming library.

3.2 Processor Registers

The 78K0R/KE3 products incorporate the following processor registers.

3.2.1 Control registers

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

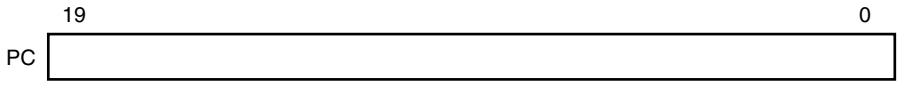
(1) Program counter (PC)

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

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

Reset signal generation sets the reset vector table values at addresses 0000H and 0001H to the program counter.

Figure 3-12. Format of Program Counter



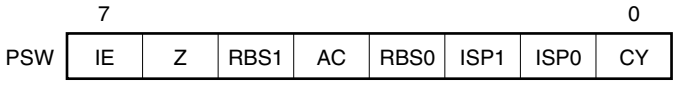
(2) Program status word (PSW)

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

Program status word contents are stored in the stack area upon interrupt request generation or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 3-13. Format of Program Status Word



(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)

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

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

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

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

(d) Auxiliary carry flag (AC)

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

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

This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 by a priority specification flag register (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see **15.3 (3)**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-14. Format of Stack Pointer

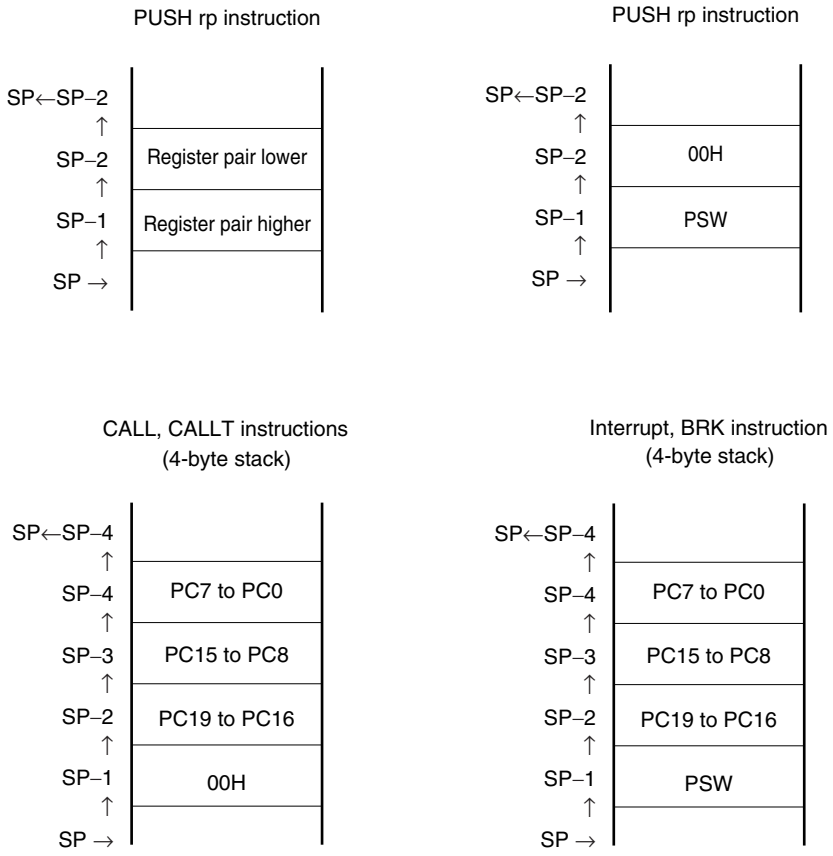
	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

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 3-15.

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

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



3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FFEE0H to FFEFFH) of the data memory. The general-purpose registers 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).

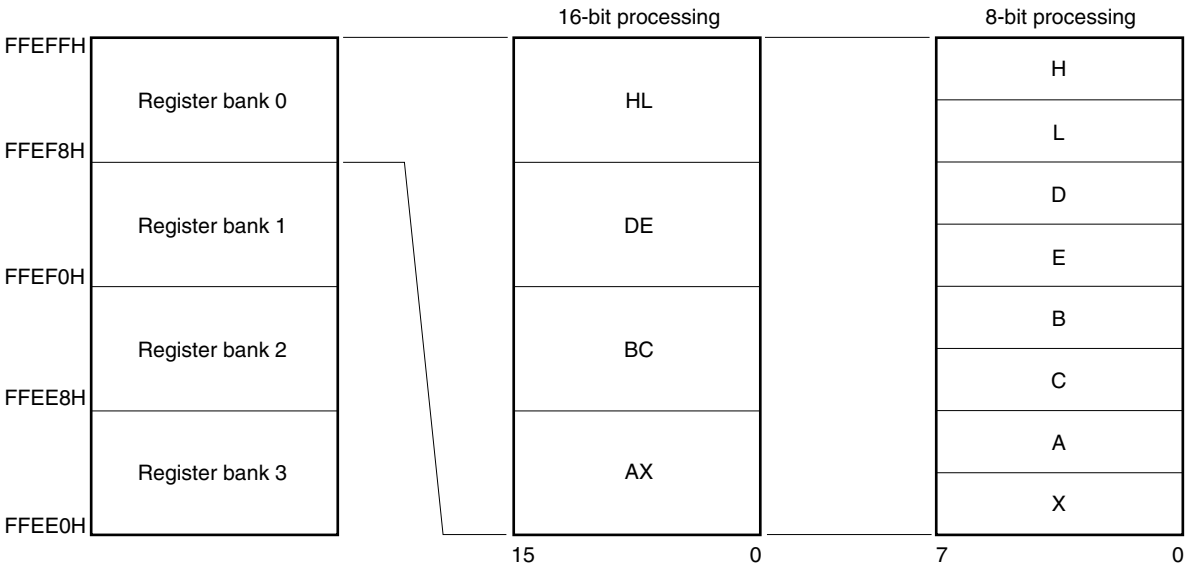
These registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

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.

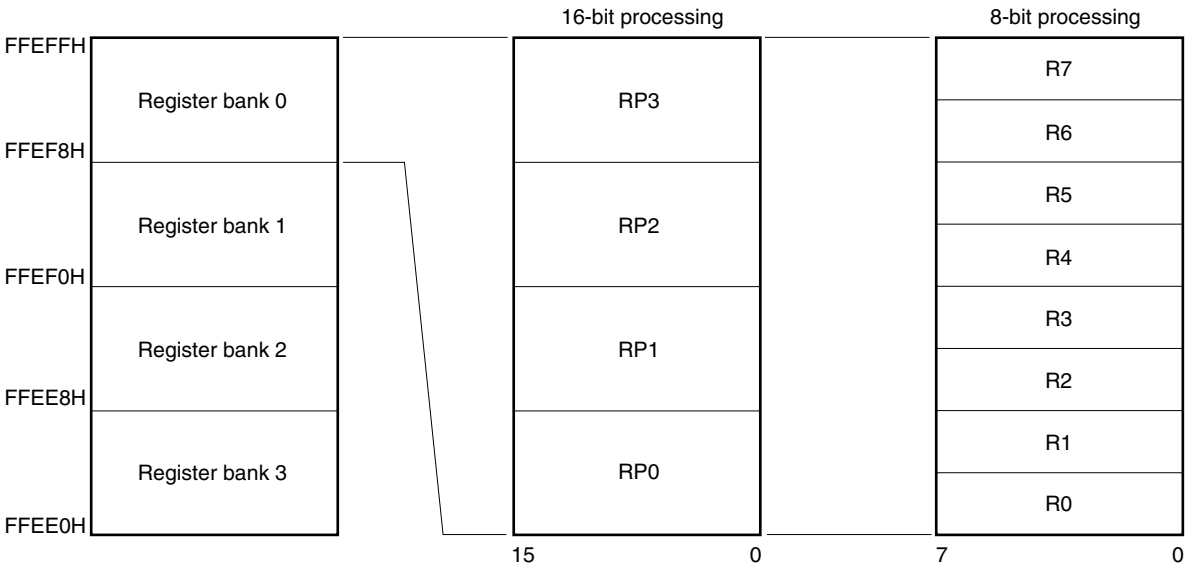
Caution It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

<R> **Figure 3-16. 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 3-17. 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 extended SFRs are not assigned.

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

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

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
FFF0CH	Port register 12	P12		R/W	√	√	–	00H
FFF0DH	Port register 13	P13		R/W	√	√	–	00H
FFF0EH	Port register 14	P14		R/W	√	√	–	00H
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	–	√	√	0000H
FFF11H		–			–	–		
FFF12H	Serial data register 01	RXD0	SDR01	R/W	–	√	√	0000H
FFF13H		–			–	–		
FFF14H	Serial data register 12	TXD3	SDR12	R/W	–	√	√	0000H
FFF15H		–			–	–		
FFF16H	Serial data register 13	RXD3	SDR13	R/W	–	√	√	0000H
FFF17H		–			–	–		
FFF18H	Timer data register 00	TDR00		R/W	–	–	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01		R/W	–	–	√	0000H
FFF1BH								
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
FFF2CH	Port mode register 12	PM12		R/W	√	√	–	FFH
FFF2EH	Port mode register 14	PM14		R/W	√	√	–	FFH

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF30H	A/D converter mode register	ADM		R/W	√	√	–	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	–	00H
FFF37H	Key return mode register	KRM		R/W	√	√	–	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	–	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	–	00H
FFF3AH	External interrupt rising edge enable register 1	EGP1		R/W	√	√	–	00H
FFF3BH	External interrupt falling edge enable register 1	EGN1		R/W	√	√	–	00H
FFF3CH	Input switch control register	ISC		R/W	√	√	–	00H
FFF3EH	Timer input select register 0	TIS0		R/W	√	√	–	00H
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	–	√	√	0000H
FFF45H		–			–	–		
FFF46H	Serial data register 03	RXD1	SDR03	R/W	–	√	√	0000H
FFF47H		–			–	–		
FFF50H	IIC shift register 0	IIC0		R/W	–	√	–	00H
FFF51H	IIC flag register 0	IICF0		R/W	√	√	–	00H
FFF52H	IIC control register 0	IICC0		R/W	√	√	–	00H
FFF53H	IIC slave address register 0	SVA0		R/W	–	√	–	00H
FFF54H	IIC clock select register 0	IICCL0		R/W	√	√	–	00H
FFF55H	IIC function expansion register 0	IICX0		R/W	√	√	–	00H
FFF56H	IIC status register 0	IICS0		R	√	√	–	00H
FFF64H	Timer data register 02	TDR02		R/W	–	–	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03		R/W	–	–	√	0000H
FFF67H								
FFF68H	Timer data register 04	TDR04		R/W	–	–	√	0000H
FFF69H								
FFF6AH	Timer data register 05	TDR05		R/W	–	–	√	0000H
FFF6BH								
FFF6CH	Timer data register 06	TDR06		R/W	–	–	√	0000H
FFF6DH								
FFF6EH	Timer data register 07	TDR07		R/W	–	–	√	0000H
FFF6FH								

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
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 1
FFF95H	Week count register	WEEK	R/W	–	√	–	00H
FFF96H	Day count register	DAY	R/W	–	√	–	01H
FFF97H	Month count register	MONTH	R/W	–	√	–	01H
FFF98H	Year count register	YEAR	R/W	–	√	–	00H
FFF99H	Watch error correction register	SUBCUD	R/W	–	√	–	00H
FFF9AH	Alarm minute register	ALARMWM	R/W	–	√	–	00H
FFF9BH	Alarm hour register	ALARMWH	R/W	–	√	–	12H
FFF9CH	Alarm week register	ALARMWW	R/W	–	√	–	00H
FFF9DH	Real-time counter control register 0	RTCC0	R/W	√	√	–	00H
FFF9EH	Real-time counter control register 1	RTCC1	R/W	√	√	–	00H
FFF9FH	Real-time counter control register 2	RTCC2	R/W	√	√	–	00H
FFFA0H	Clock operation mode control register	CMC	R/W	–	√	–	00H
FFFA1H	Clock operation status control register	CSC	R/W	√	√	–	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC	R	√	√	–	00H
FFFA3H	Oscillation stabilization time select register	OSTS	R/W	–	√	–	07H
FFFA4H	System clock control register	CKC	R/W	√	√	–	09H
FFFA5H	Clock output select register 0	CKS0	R/W	√	√	–	00H
FFFA6H	Clock output select register 1	CKS1	R/W	√	√	–	00H
FFFA8H	Reset control flag register	RESF	R	–	√	–	00H Note 2
FFFA9H	Low-voltage detection register	LVIM	R/W	√	√	–	00H Note 3
FFFAAH	Low-voltage detection level select register	LVIS	R/W	√	√	–	0EH Note 4
FFFABH	Watchdog timer enable register	WDTE	R/W	–	√	–	1A/9A Note 5
<R> FFFACH	–	TTBLH Note 6	–	–	–	–	Undefined
FFFADH							
<R> FFFAEH	–	TTBLL Note 6	–	–	–	–	Undefined
FFFAFH							

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

2. The reset value of RESF varies depending on the reset source.

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

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

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

<R> **6.** This SFR cannot be used by the user, so do not operate it directly.

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

	Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
						1-bit	8-bit	16-bit	
	FFFB0H	DMA SFR address register 0	DSA0		R/W	–	√	–	00H
	FFFB1H	DMA SFR address register 1	DSA1		R/W	–	√	–	00H
	FFFB2H	DMA RAM address register 0L	DRA0L	DRA0	R/W	–	√	√	00H
	FFFB3H	DMA RAM address register 0H	DRA0H		R/W	–	√		00H
	FFFB4H	DMA RAM address register 1L	DRA1L	DRA1	R/W	–	√	√	00H
	FFFB5H	DMA RAM address register 1H	DRA1H		R/W	–	√		00H
	FFFB6H	DMA byte count register 0L	DBC0L	DBC0	R/W	–	√	√	00H
	FFFB7H	DMA byte count register 0H	DBC0H		R/W	–	√		00H
	FFFB8H	DMA byte count register 1L	DBC1L	DBC1	R/W	–	√	√	00H
	FFFB9H	DMA byte count register 1H	DBC1H		R/W	–	√		00H
	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
<R>	FFFC0H	–	PFCMD[Note]		–	–	–	–	Undefined
<R>	FFFC2H	–	PFS[Note]		–	–	–	–	Undefined
<R>	FFFC4H	–	FLPMC[Note]		–	–	–	–	Undefined
	FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	√	√	√	00H
	FFFD1H	Interrupt request flag register 2H	IF2H		R/W	√	√		00H
	FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH
	FFFD5H	Interrupt mask flag register 2H	MK2H		R/W	√	√		FFH
	FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH
	FFFD9H	Priority specification flag register 02H	PR02H		R/W	√	√		FFH
	FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
	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

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFEEH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	Priority specification flag register 11H	PR11H			√	√		FFH
FFFF0H	Multiplication input data register A	MULA		R/W	–	–	√	0000H
FFFF1H								
FFFF2H	Multiplication input data register B	MULB		R/W	–	–	√	0000H
FFFF3H								
FFFF4H	Higher multiplication result storage register	MULOH		R	–	–	√	0000H
FFFF5H								
FFFF6H	Lower multiplication result storage register	MULOL		R	–	–	√	0000H
FFFF7H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	–	00H

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 extended SFRs are not assigned.

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

Table 3-6. Extended SFR (2nd SFR) List (1/4)

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
F0037H	Pull-up resistor option register 7	PU7		R/W	√	√	–	00H
F003CH	Pull-up resistor option register 12	PU12		R/W	√	√	–	00H
F003EH	Pull-up resistor option register 14	PU14		R/W	√	√	–	00H
F0040H	Port input mode register 0	PIM0		R/W	√	√	–	00H
F0050H	Port output mode register 0	POM0		R/W	√	√	–	00H
F0060H	Noise filter enable register 0	NFEN0		R/W	√	√	–	00H
F0061H	Noise filter enable register 1	NFEN1		R/W	√	√	–	00H
F00F0H	Peripheral enable register 0	PER0		R/W	√	√	–	00H
F00F2H	Internal high-speed oscillator trimming register	HIOTRM		R/W	–	√	–	10H
F00F3H	Operation speed mode control register	OSMC		R/W	–	√	–	00H
F00F4H	Regulator mode control register	RMC		R/W	–	√	–	00H
F00FEH	BCD adjust result register	BCDADJ		R	–	√	–	Undefined
F0100H	Serial status register 00	SSR00L	SSR00	R	–	√	√	0000H
F0101H		–			–	–		
F0102H	Serial status register 01	SSR01L	SSR01	R	–	√	√	0000H
F0103H		–			–	–		
F0104H	Serial status register 02	SSR02L	SSR02	R	–	√	√	0000H
F0105H		–			–	–		
F0106H	Serial status register 03	SSR03L	SSR03	R	–	√	√	0000H
F0107H		–			–	–		
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	–	√	√	0000H
F0109H		–			–	–		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	–	√	√	0000H
F010BH		–			–	–		
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								

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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		–			–	–		
F0144H	Serial status register 12	SSR12L	SSR12	R	–	√	√	0000H
F0145H		–			–	–		
F0146H	Serial status register 13	SSR13L	SSR13	R	–	√	√	0000H
F0147H		–			–	–		
F014CH	Serial flag clear trigger register 12	SIR12L	SIR12	R/W	–	√	√	0000H
F014DH		–			–	–		
F014EH	Serial flag clear trigger register 13	SIR13L	SIR13	R/W	–	√	√	0000H
F014FH		–			–	–		
F0154H	Serial mode register 12	SMR12		R/W	–	–	√	0020H
F0155H								
F0156H	Serial mode register 13	SMR13		R/W	–	–	√	0020H
F0157H								
F015CH	Serial communication operation setting register 12	SCR12		R/W	–	–	√	0087H
F015DH								
F015EH	Serial communication operation setting register 13	SCR13		R/W	–	–	√	0087H
F015FH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	√	√	√	0000H
F0161H		–			–	–		
F0162H	Serial channel start register 1	SS1L	SS1	R/W	√	√	√	0000H
F0163H		–			–	–		

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

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01A0H	Timer status register 00	TSR00L	TSR00	R	–	√	√	0000H
F01A1H		–			–	–		
F01A2H	Timer status register 01	TSR01L	TSR01	R	–	√	√	0000H
F01A3H		–			–	–		
F01A4H	Timer status register 02	TSR02L	TSR02	R	–	√	√	0000H
F01A5H		–			–	–		
F01A6H	Timer status register 03	TSR03L	TSR03	R	–	√	√	0000H
F01A7H		–			–	–		
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		–			–	–		

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

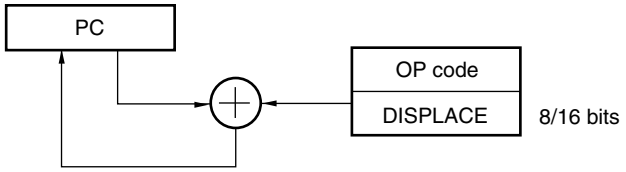
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

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

Figure 3-18. Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

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

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

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

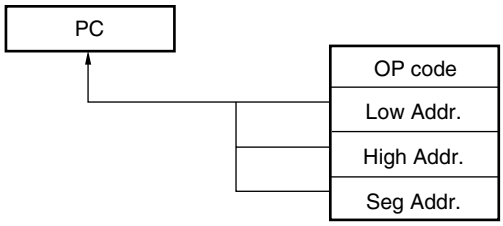
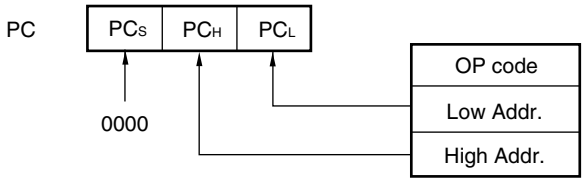


Figure 3-20. Example of CALL !addr16/BR !addr16



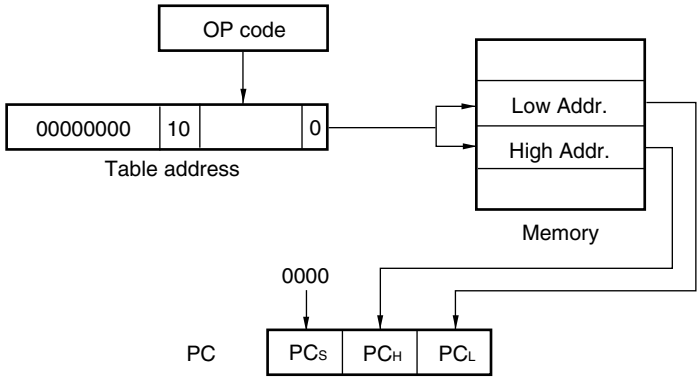
3.3.3 Table indirect addressing

[Function]

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

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

Figure 3-21. Outline of Table Indirect Addressing

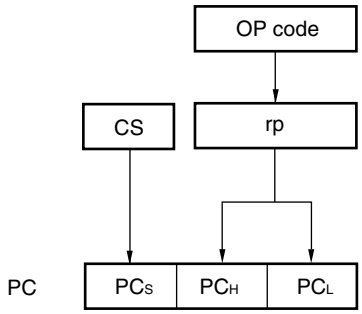


3.3.4 Register direct addressing

[Function]

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

Figure 3-22. Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

[Function]

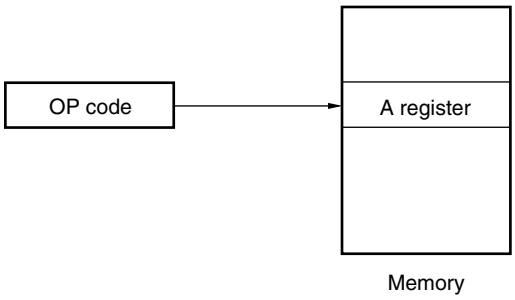
Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[Operand format]

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

Implied addressing can be applied only to MULU X.

Figure 3-23. Outline of Implied Addressing



3.4.2 Register addressing

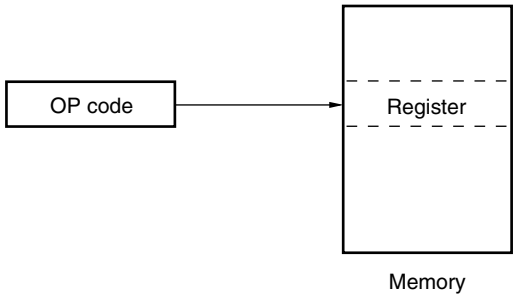
[Function]

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

[Operand format]

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

Figure 3-24. Outline of Register Addressing



3.4.3 Direct addressing

[Function]

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

[Operand format]

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

Figure 3-25. Example of ADDR16

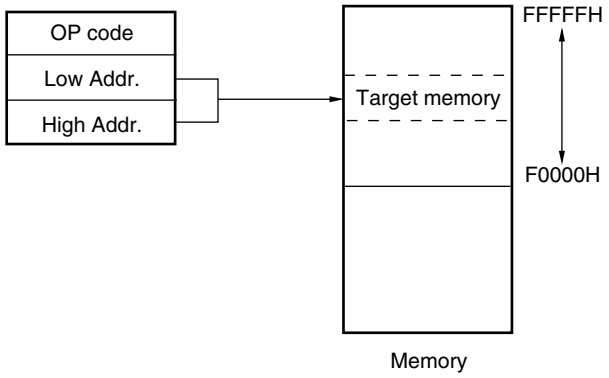
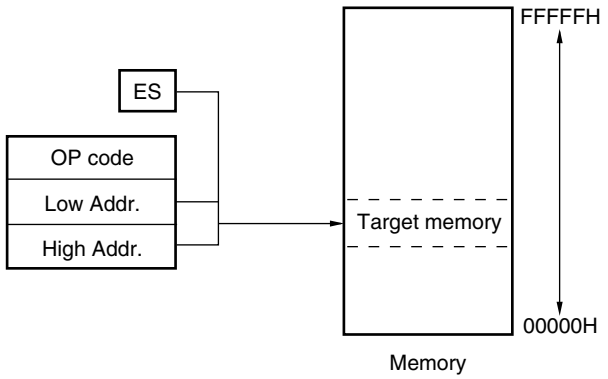


Figure 3-26. Example of ES:ADDR16



3.4.4 Short direct addressing

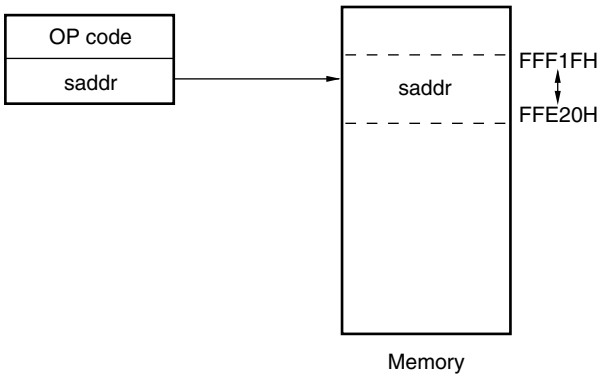
[Function]

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

[Operand format]

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

Figure 3-27. Outline of Short Direct Addressing



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

Regardless of whether SADDR or SADDRP is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

3.4.5 SFR addressing

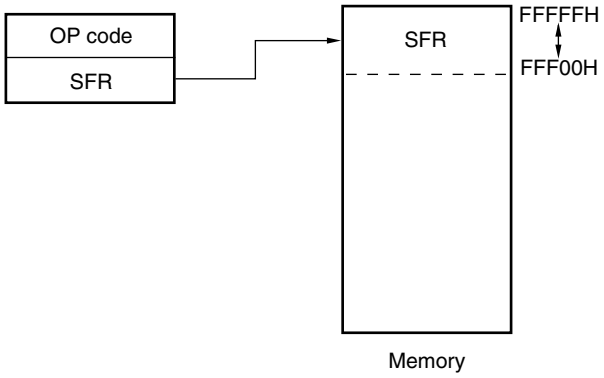
[Function]

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

[Operand format]

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

Figure 3-28. Outline of SFR Addressing



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

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

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

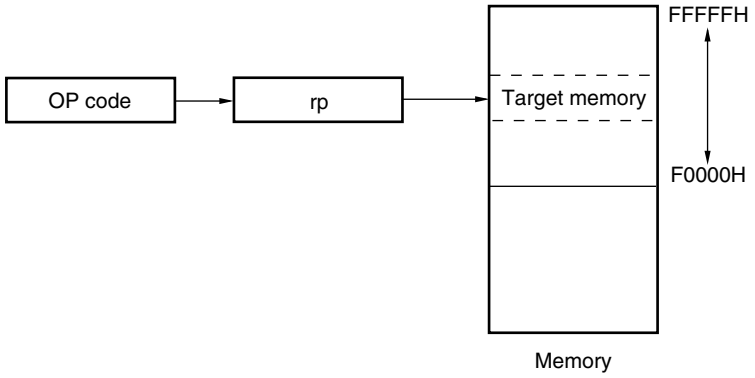
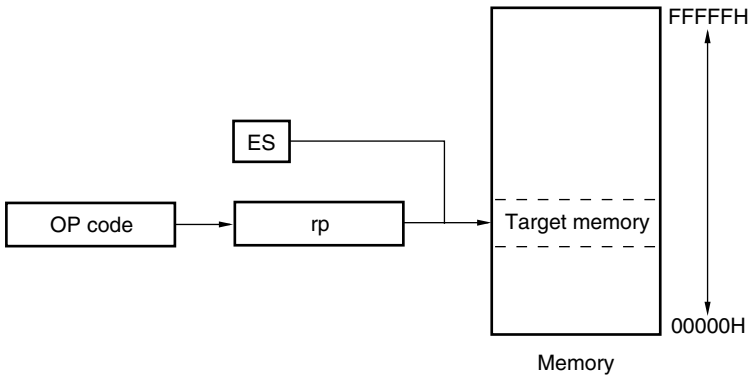


Figure 3-30. Example of ES:[DE], ES:[HL]



3.4.7 Based addressing

[Function]

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

[Operand format]

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

Figure 3-31. Example of [SP+byte]

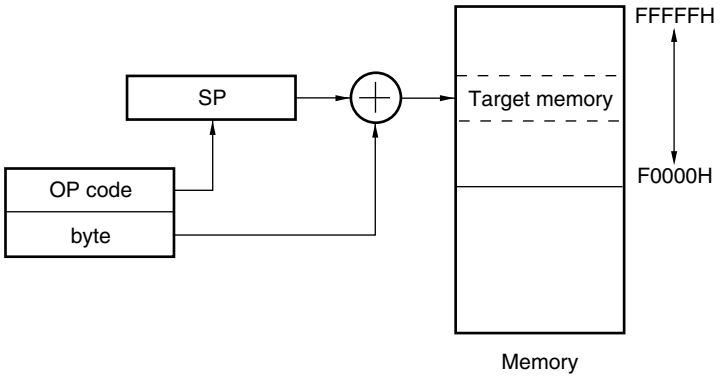


Figure 3-32. Example of [HL + byte], [DE + byte]

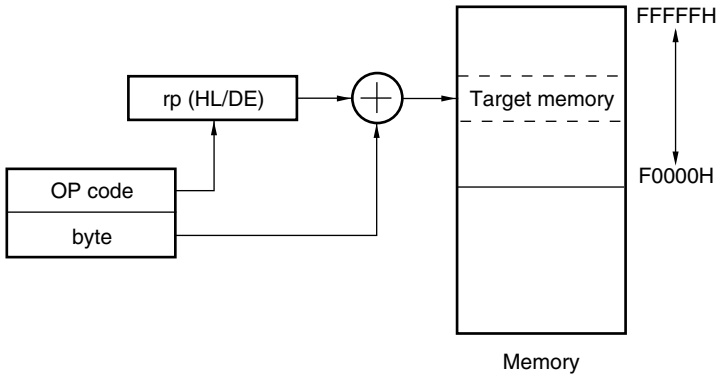


Figure 3-33. Example of word[B], word[C]

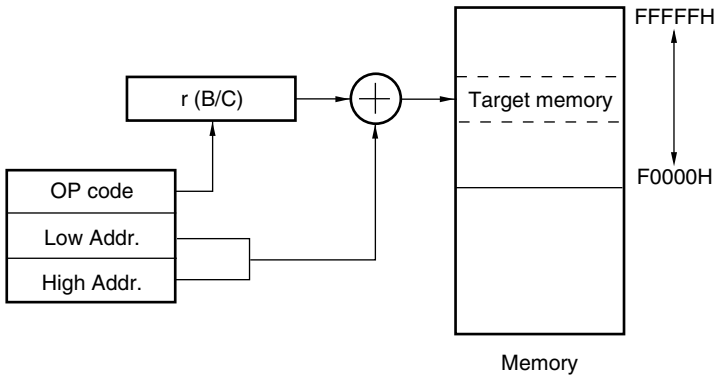


Figure 3-34. Example of word[BC]

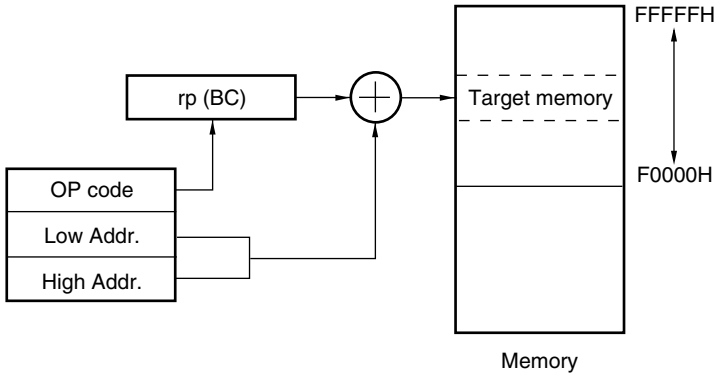


Figure 3-35. Example of ES:[HL + byte], ES:[DE + byte]

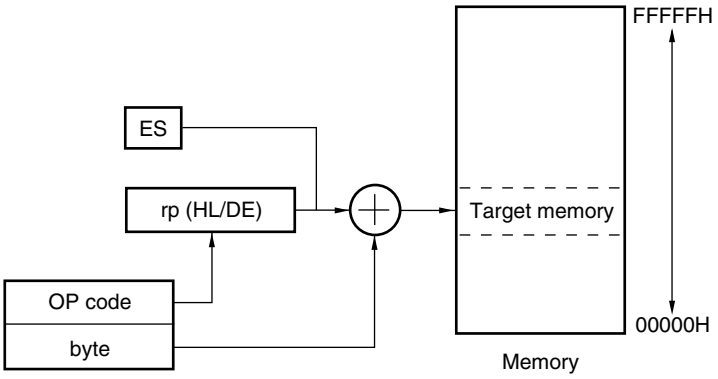


Figure 3-36. Example of ES:word[B], ES:word[C]

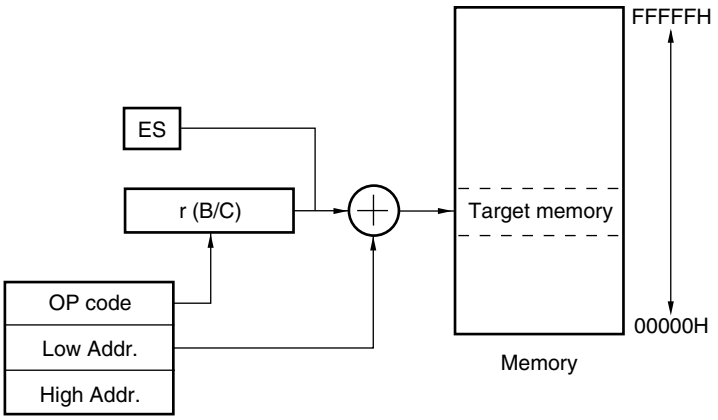
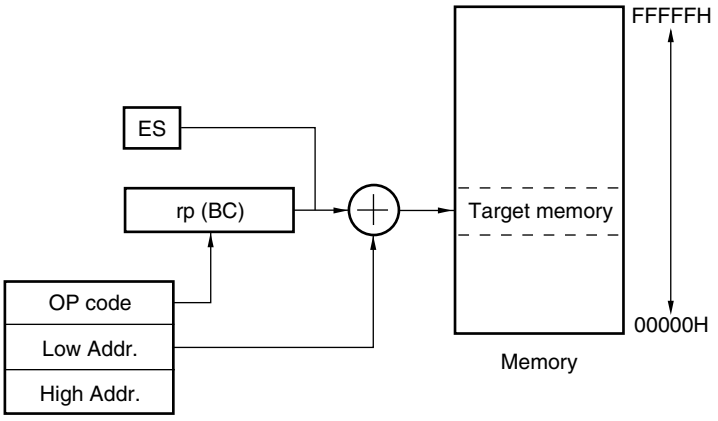


Figure 3-37. Example of ES:word[BC]



3.4.8 Based indexed addressing

[Function]

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

[Operand format]

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

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

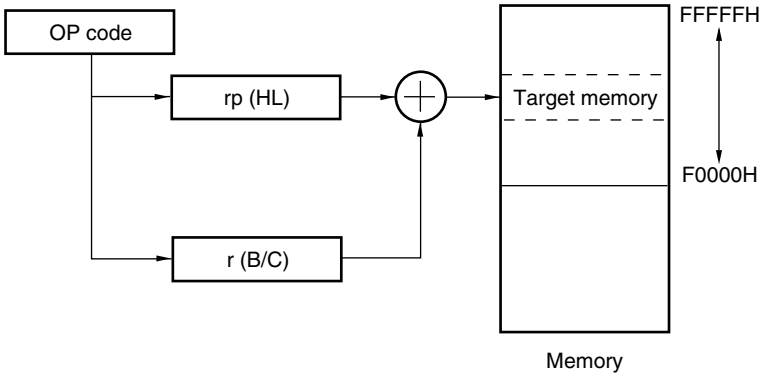
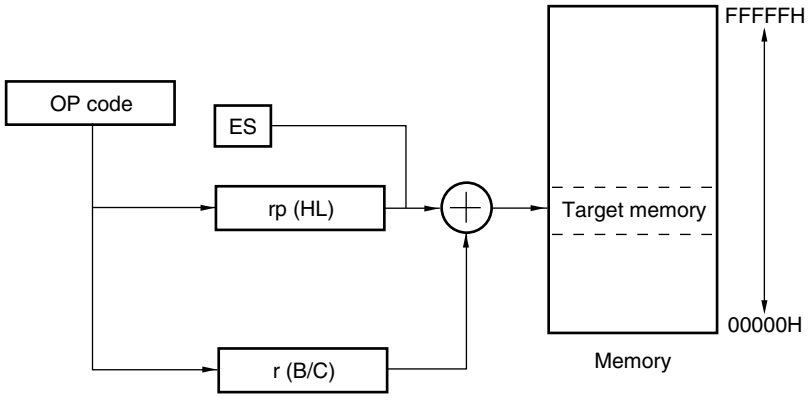


Figure 3-39. Example of ES:[HL+B], ES:[HL+C]



3.4.9 Stack addressing

[Function]

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

Stack addressing is applied only to the internal RAM area.

[Operand format]

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

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

There are three types of pin I/O buffer power supplies: AV_{REF}, EV_{DD}, and V_{DD}. The relationship between these power supplies and the pins is shown below.

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

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

78K0R/KE3 products are provided with the ports shown in Figure 4-1, which enable variety of control operations. The functions of each port are shown in Table 4-2.

In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, see **CHAPTER 2 PIN FUNCTIONS**.

Figure 4-1. Port Types

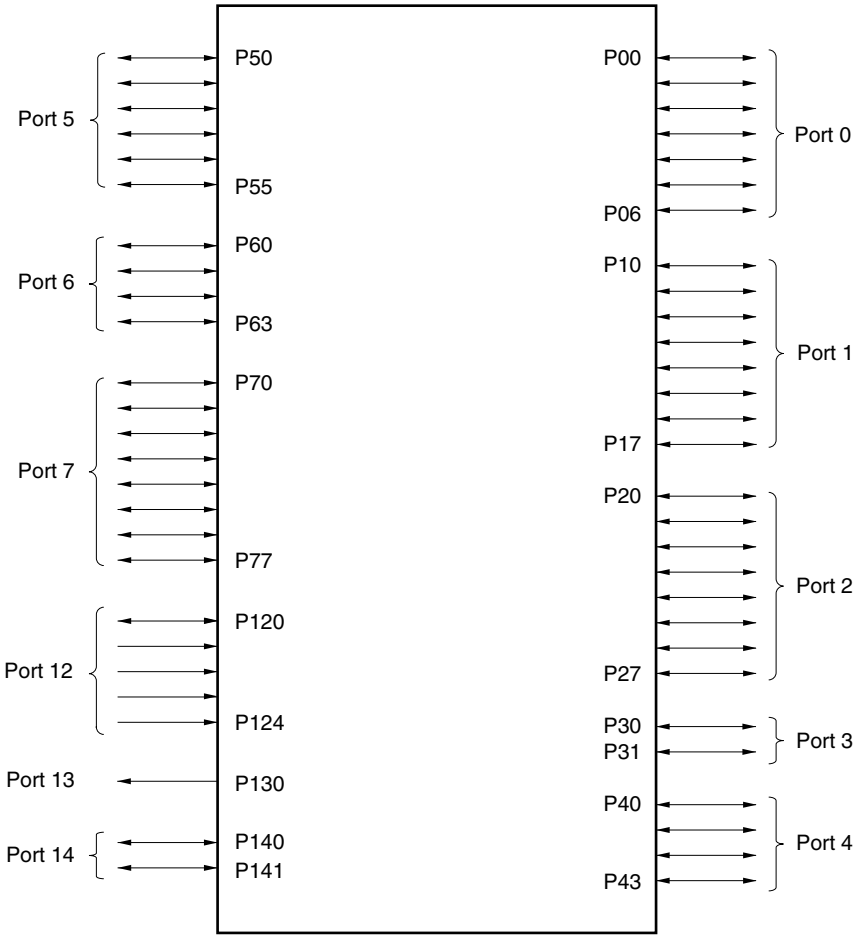


Table 4-2. Port Functions (1/2)

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				TI05/TO05
P06				TI06/TO06
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{\text{SCK00}}$
P11				SI00/RxD0
P12				SO00/TxD0
P13				TxD3
P14				RxD3
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. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOOL0
P41				TOOL1
P42				TI04/TO04
P43				–
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				–
P53				–
P54				–
P55				–

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 P43 (port 4)**).

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

Function Name	I/O	Function	After Reset	Alternate Function
P60	I/O	Port 6. 4-bit I/O port. Output of P60 to P63 can be set to N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCL0
P61				SDA0
P62				–
P63				–
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 to P77				KR4/INTP8 to KR7/INTP11
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. 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	PCLBUZ0/INTP6
P141				PCLBUZ1/INTP7

4.2 Port Configuration

Ports include the following hardware.

Table 4-3. Port Configuration

Item	Configuration
Control registers	Port mode registers (PM0 to PM7, PM12, PM14) Port registers (P0 to P7, P12 to P14) Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU7, PU12, PU14) Port input mode registers (PIM0) Port output mode registers (POM0) A/D port configuration register (ADPC)
Port	Total: 55 (CMOS I/O: 46, CMOS input: 4, CMOS output: 1, N-ch open drain I/O: 4)
Pull-up resistor	Total: 38

4.2.1 Port 0

Port 0 is a 7-bit I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to 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.

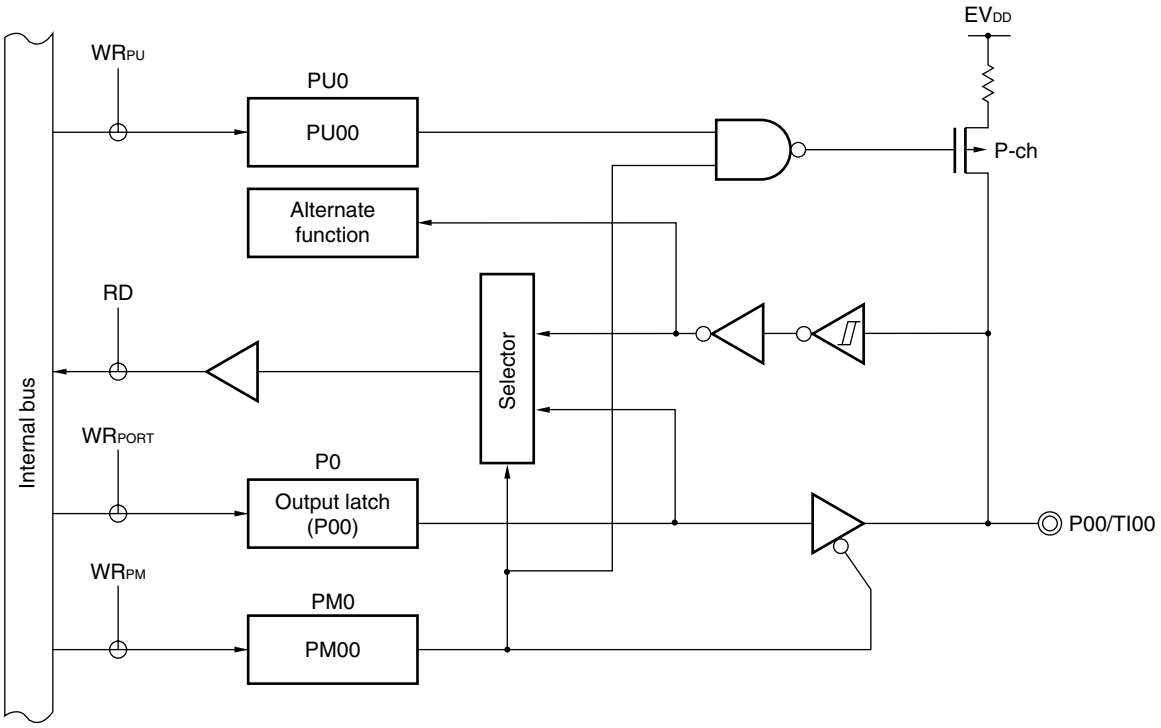
Reset signal generation sets port 0 to input mode.

Figures 4-2 to 4-6 show block diagrams of port 0.

<R> **Cautions 1. To use P01/TO00, P05/TI05/TO05, or P06/TI06/TO06 as a general-purpose port, set bits 0, 5, and 6 (TO00, TO05, TO06) of timer output register 0 (TO0) and bits 0, 5, and 6 (TOE00, TOE05, TOE06) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.**

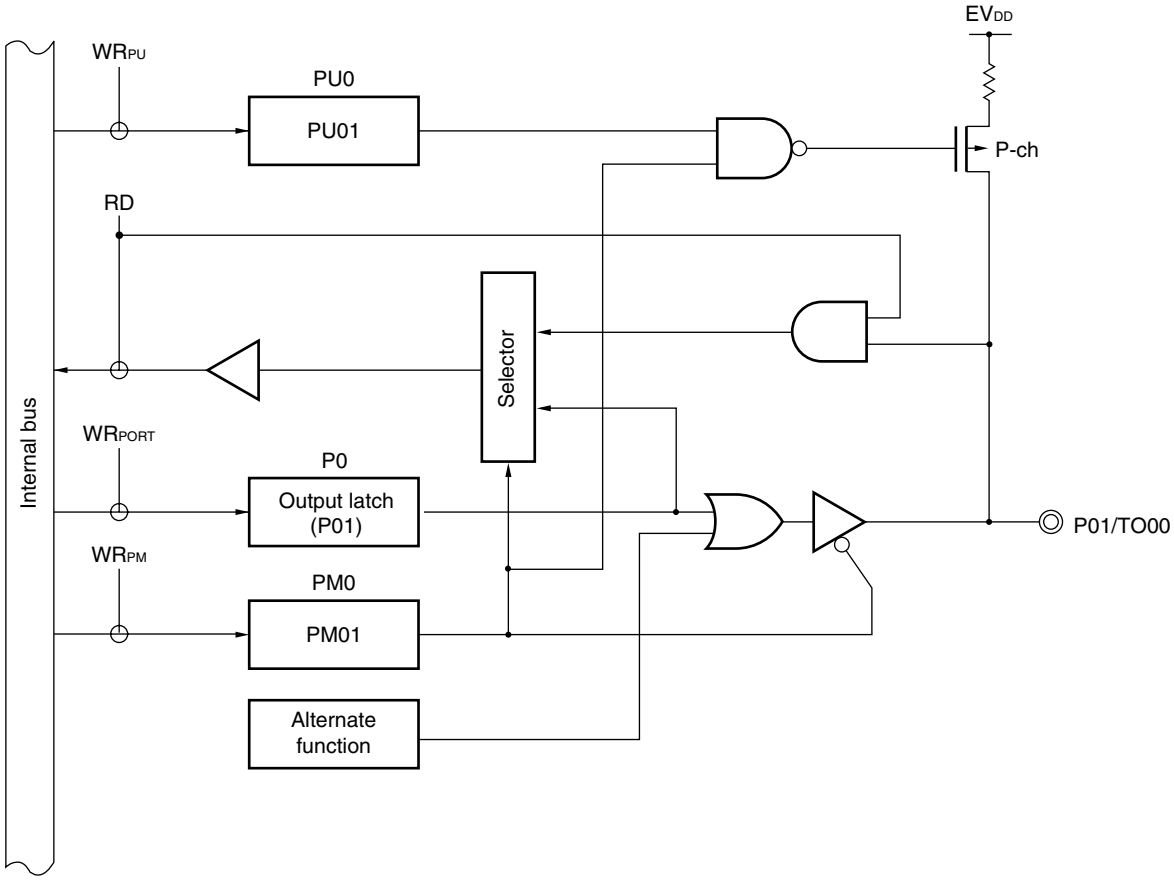
<R> **2. To use P02/SO10/TxD1, P03/SI10/RxD1/SDA10 or P04/$\overline{\text{SCK10}}$/SCL10 as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 11-7 Relationship Between Register Settings and Pins (Channel 2 of Unit 0: CSI10, UART1 Transmission, IIC10) and Table 11-8 Relationship Between Register Settings and Pins (Channel 3 of Unit 0: UART1 reception).**

Figure 4-2. Block Diagram of P00



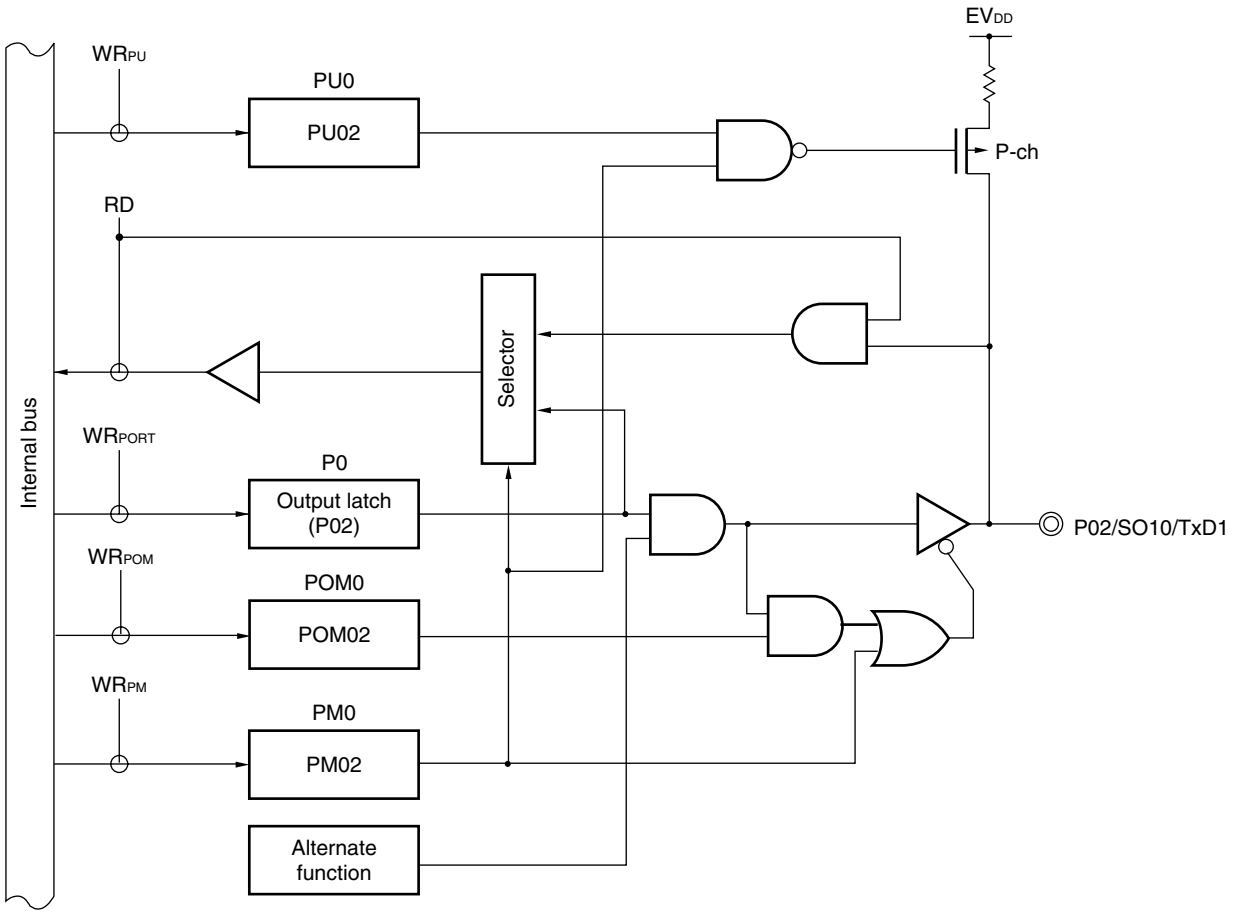
P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
RD: Read signal
WR××: Write signal

Figure 4-3. Block Diagram of P01



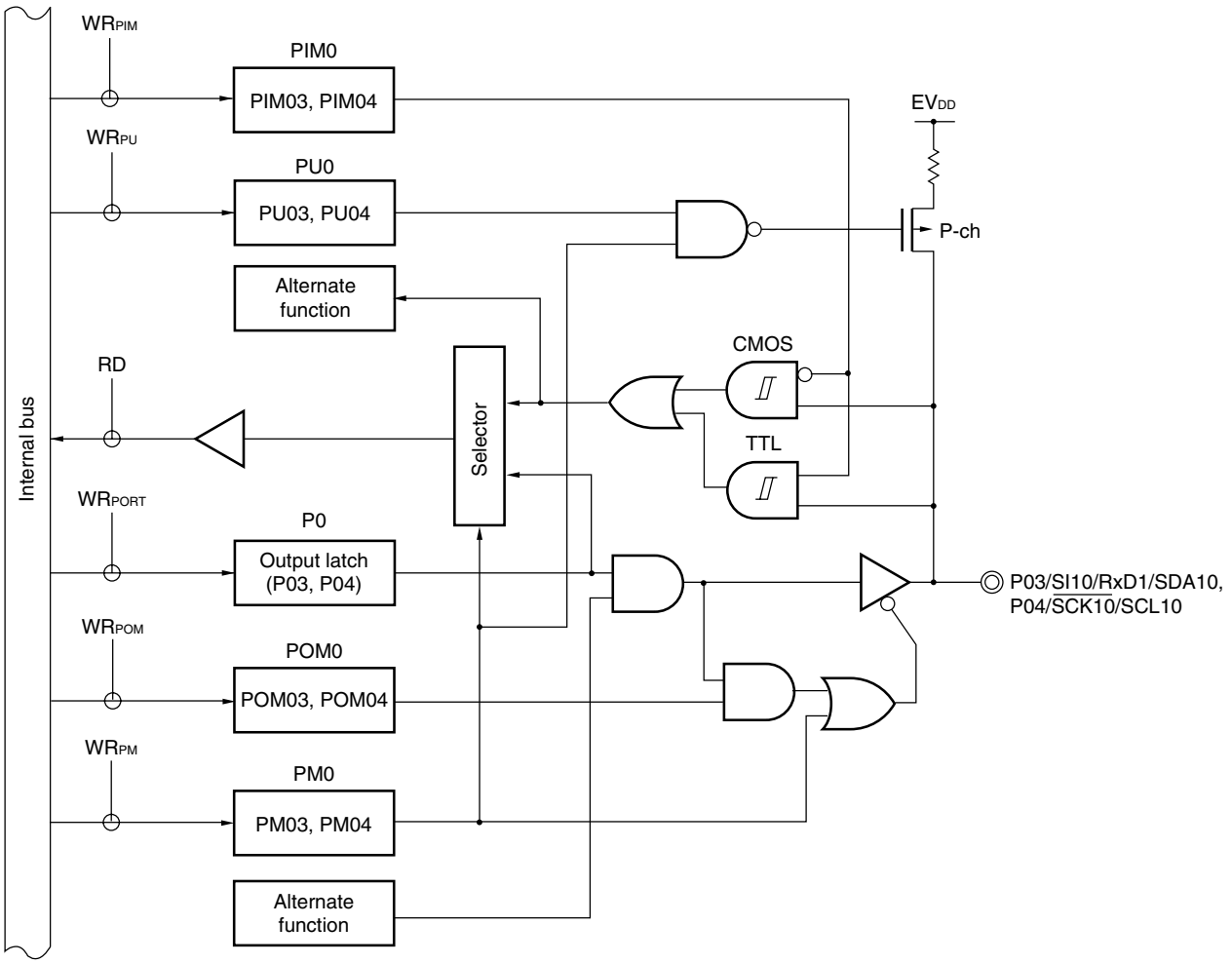
P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
RD: Read signal
WR××: Write signal

<R> **Figure 4-4. Block Diagram of P02**



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

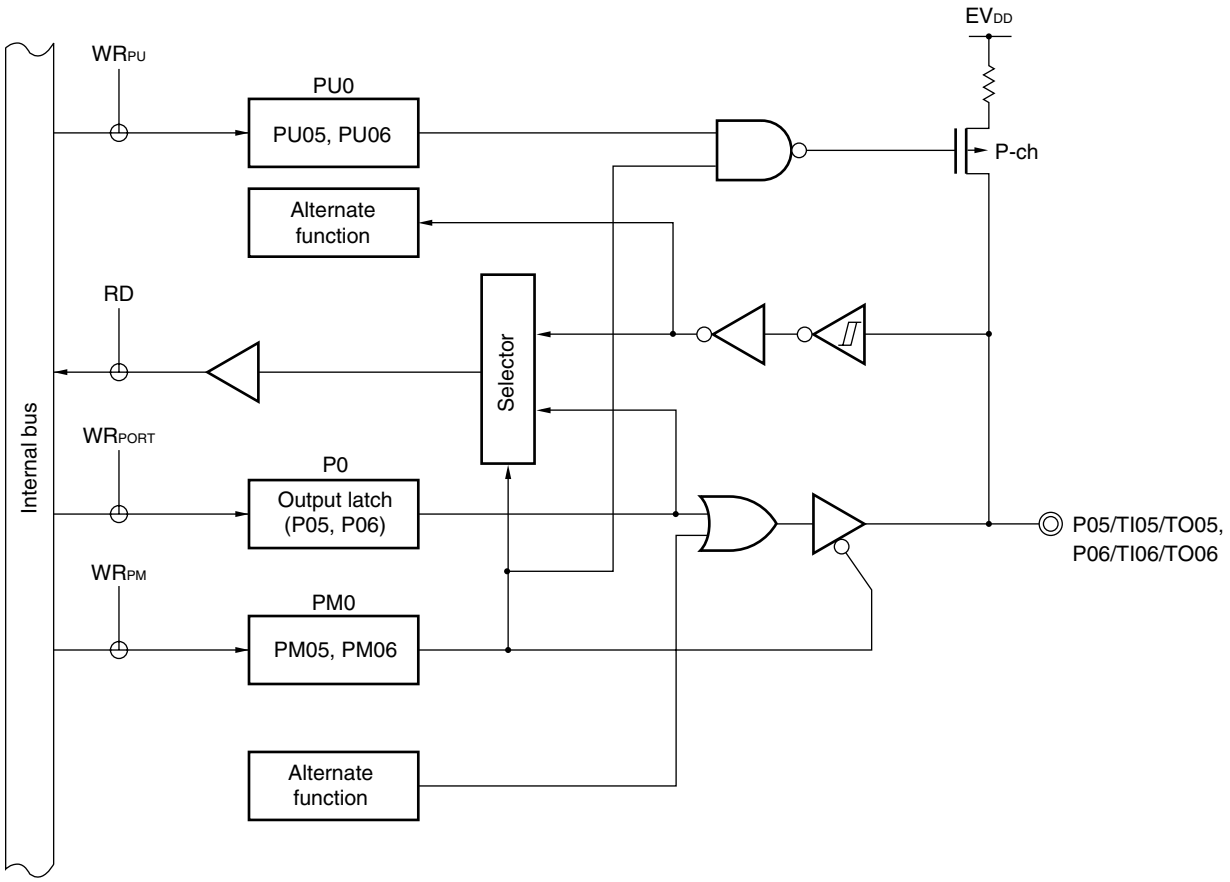
<R> **Figure 4-5. Block Diagram of P03 and P04**



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

Figure 4-6. Block Diagram of P05 and P06

<R>



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).

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.

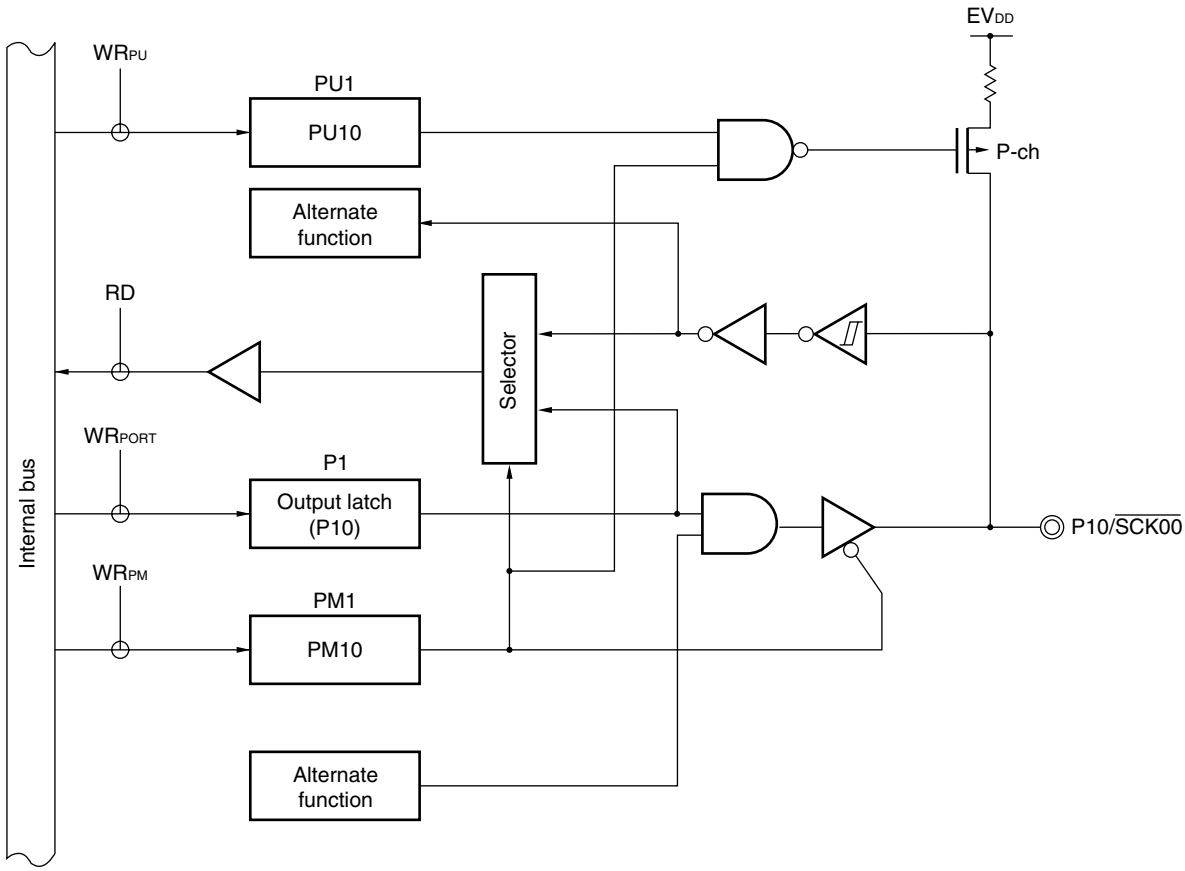
Figures 4-7 to 4-11 show block diagrams of port 1.

<R> **Cautions 1. To use P10/$\overline{\text{SCK00}}$, P11/SI00/RxD0, P12/SO00/TxD0, P13/TxD3 or P14/RxD3 as a general-purpose port, note the serial array unit setting. For details, refer to 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: UART0 Reception) , Table 11-9 Relationship Between Register Settings and Pins (Channel 2 of Unit 1: UART3 Transmission), and Table 11-10 Relationship Between Register Settings and Pins (Channel 3 of Unit 1: UART3 Reception).**

<R> **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.**

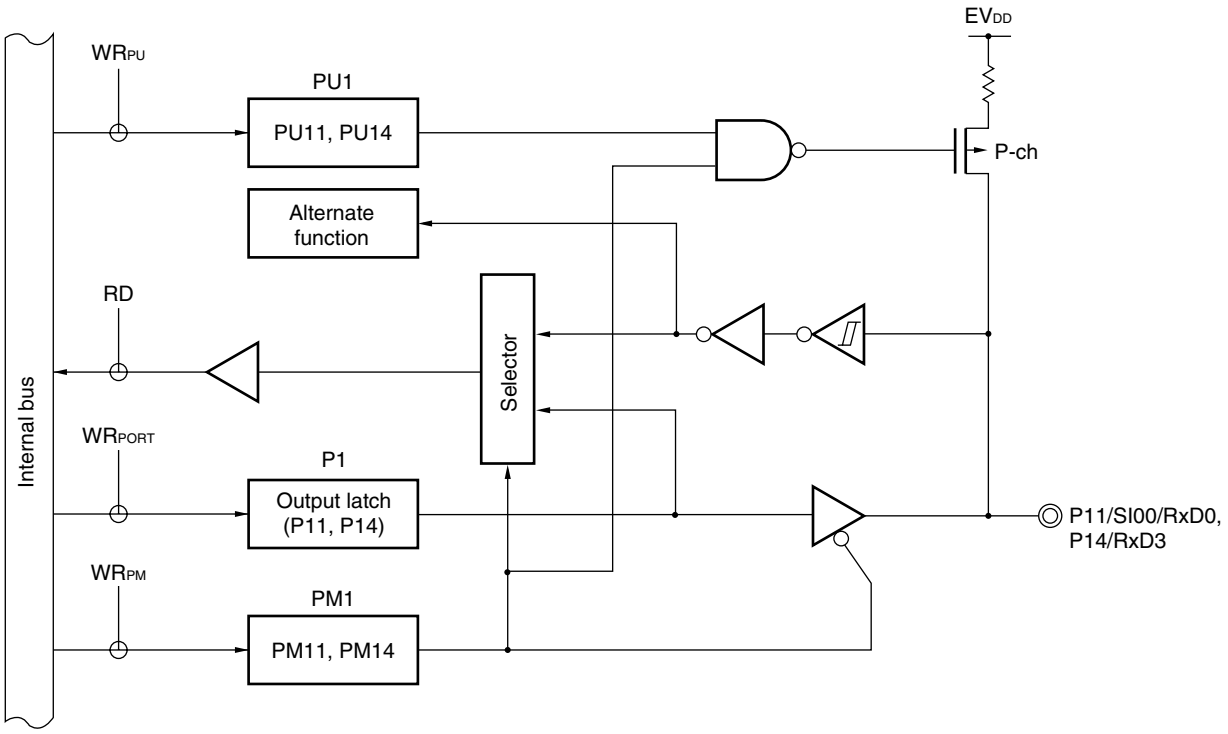
<R> **3. To use P15/RTCDIV/RTCCL as a general-purpose port, set bit 4 (RCLOE0) of Real-time counter control register 0 (RTCC0) and bit 6 (RCLOE2) of Real-time counter control register 2 (RTCC2) to "0", which is the same as their default status setting.**

Figure 4-7. Block Diagram of P10



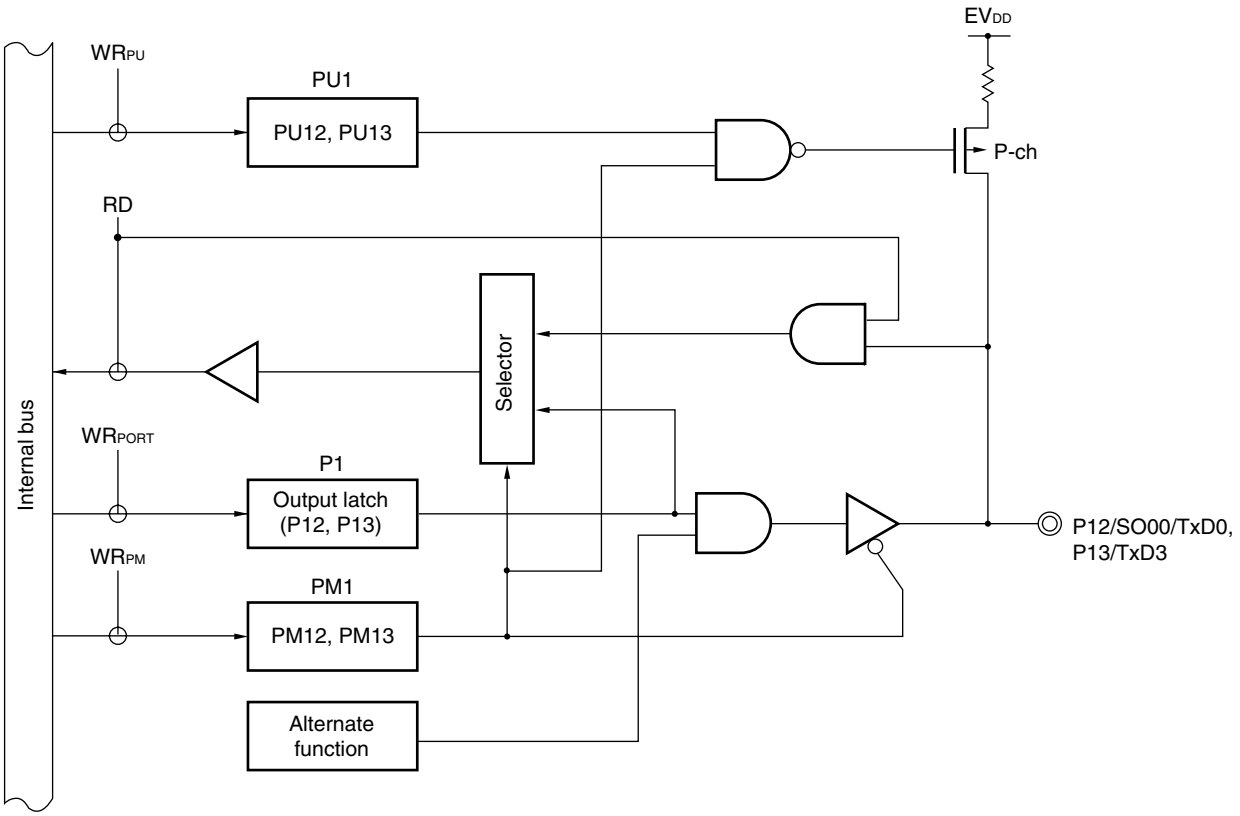
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-8. Block Diagram of P11 and P14



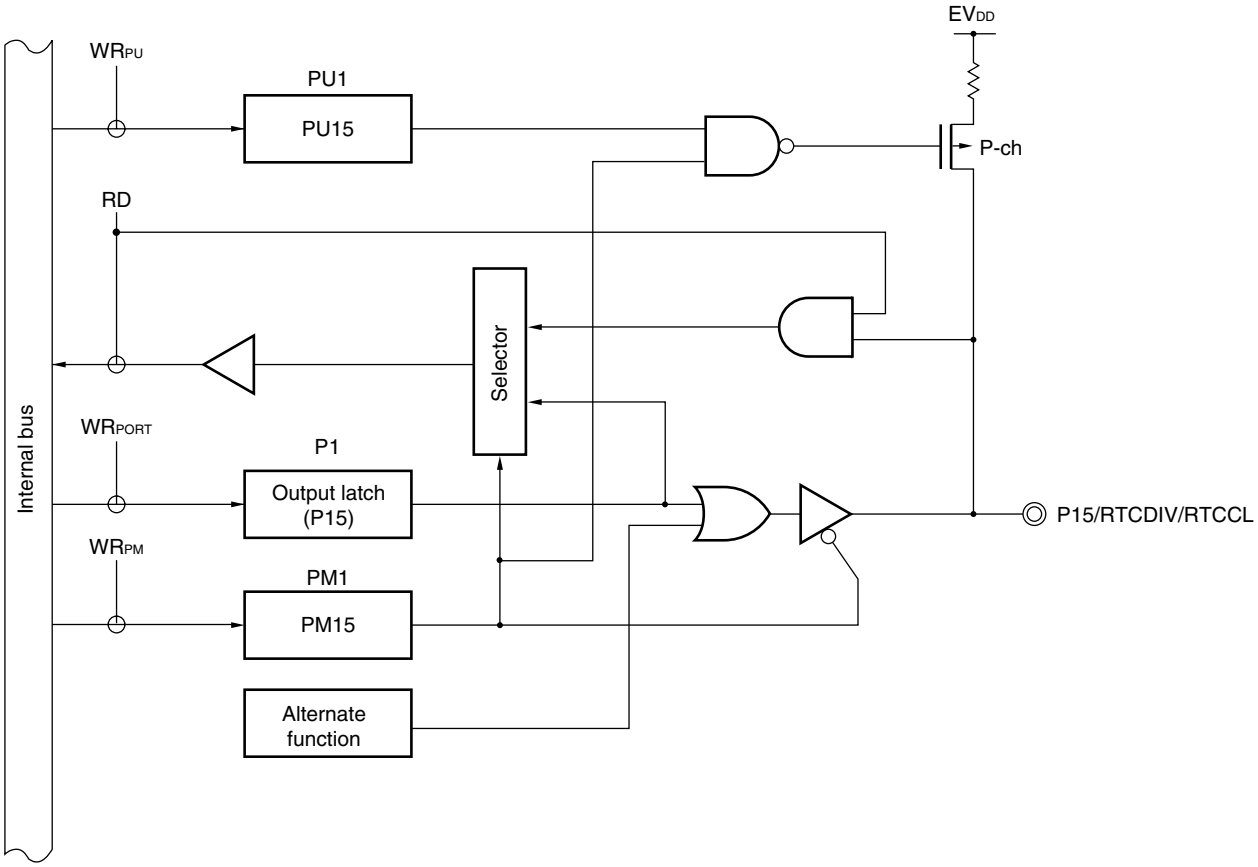
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-9. Block Diagram of P12 and P13



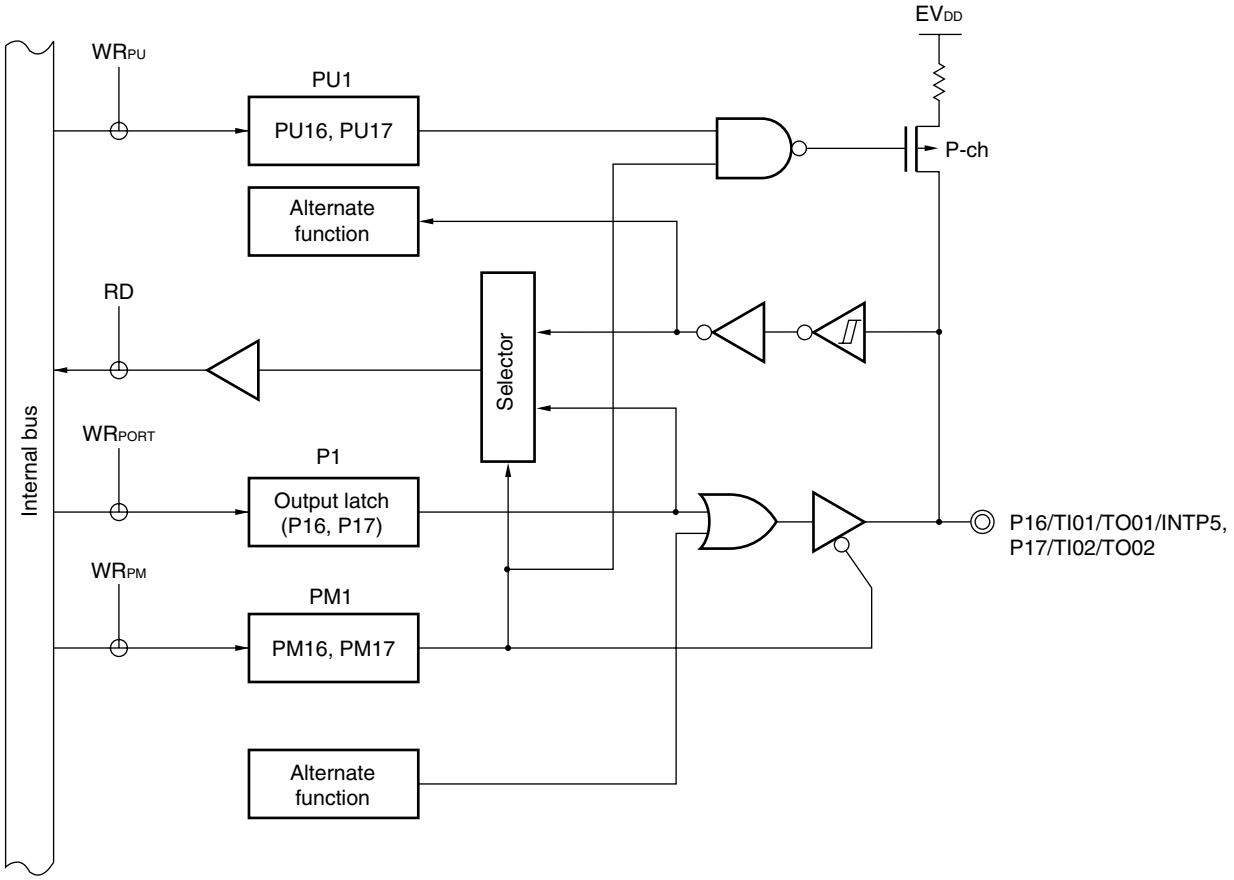
P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-10. 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

Figure 4-11. 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

4.2.3 Port 2

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

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

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

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

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

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

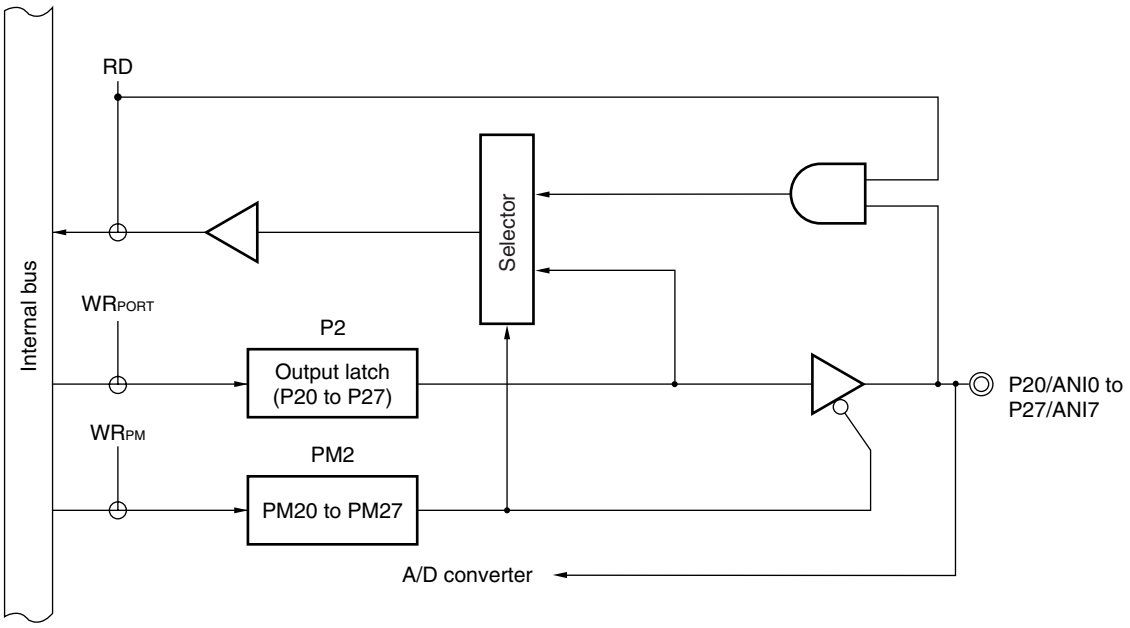
ADPC	PM2	ADS	P20/ANI0 to P27/ANI7 Pins
Digital I/O selection	Input mode	−	Digital input
	Output mode	−	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
	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

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

Figure 4-12 shows a block diagram of port 2.

Caution Make the AV_{REF} pin the same potential as the V_{DD} pin when port 2 is used as a digital port.

Figure 4-12. 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.

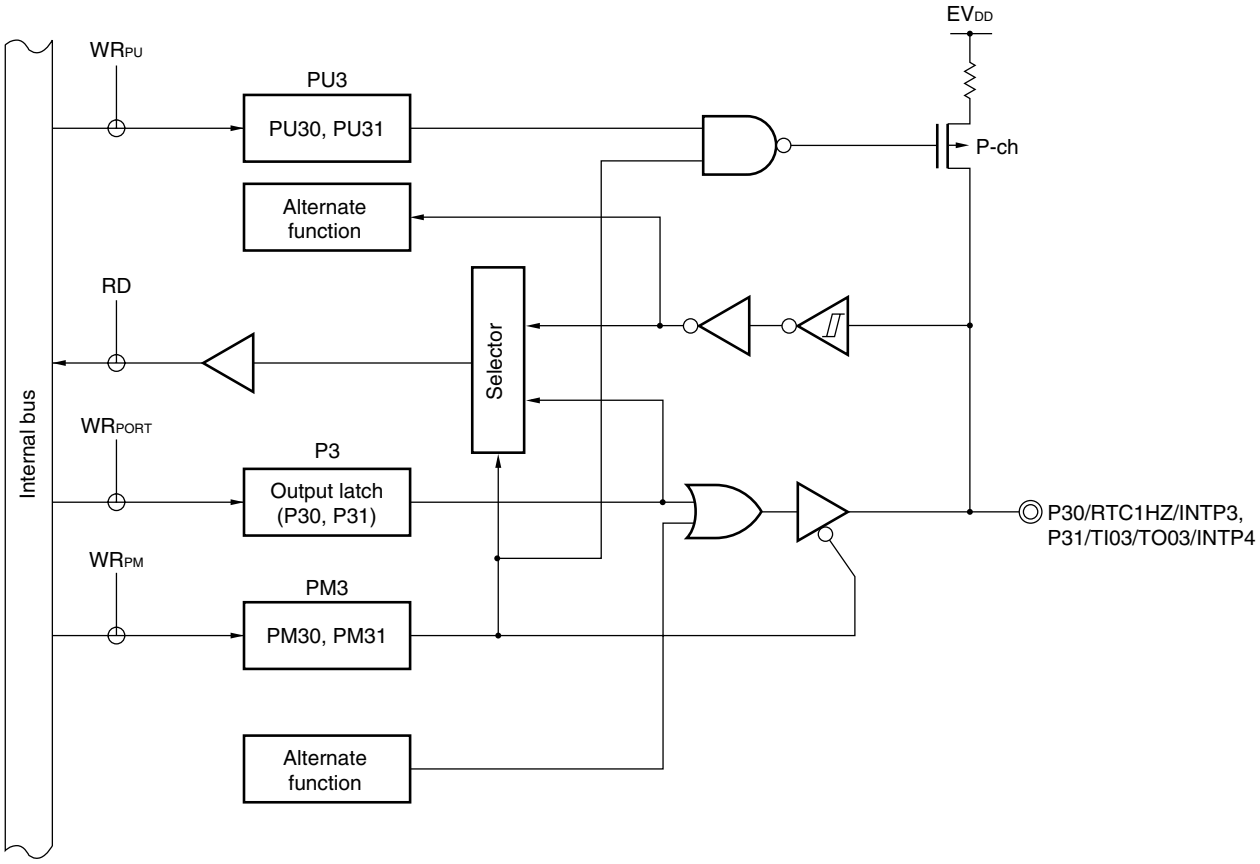
Reset signal generation sets port 3 to input mode.

Figure 4-13 shows block a diagram of port 3.

<R> **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.**

<R> **2. To use P30/RTC1HZ/INTP3 as a general-purpose port, set bit 5 (RCLOE1) of Real-time counter control register 0 (RTCC0) to "0", which is the same as their default status setting.**

Figure 4-13. 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

4.2.5 Port 4

Port 4 is an 4-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].

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

Reset signal generation sets port 4 to input mode.

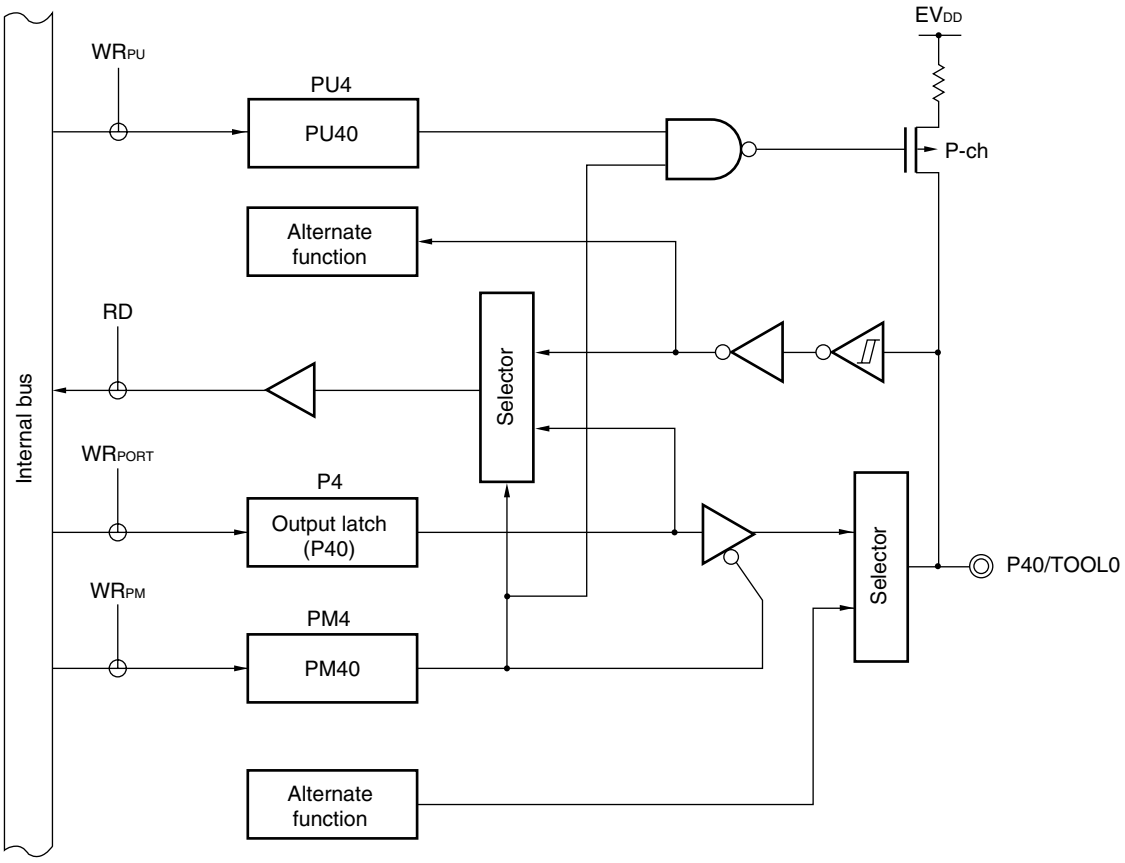
Figures 4-14 to 4-17 show block diagrams of port 4.

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

Cautions 1. When a tool is connected, the P40 pin cannot be used as a port pin.
When the on-chip debug function is used, P41 pin can be used as follows by the mode setting on the debugger.
1-line mode: can be used as a port (P41).
2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).

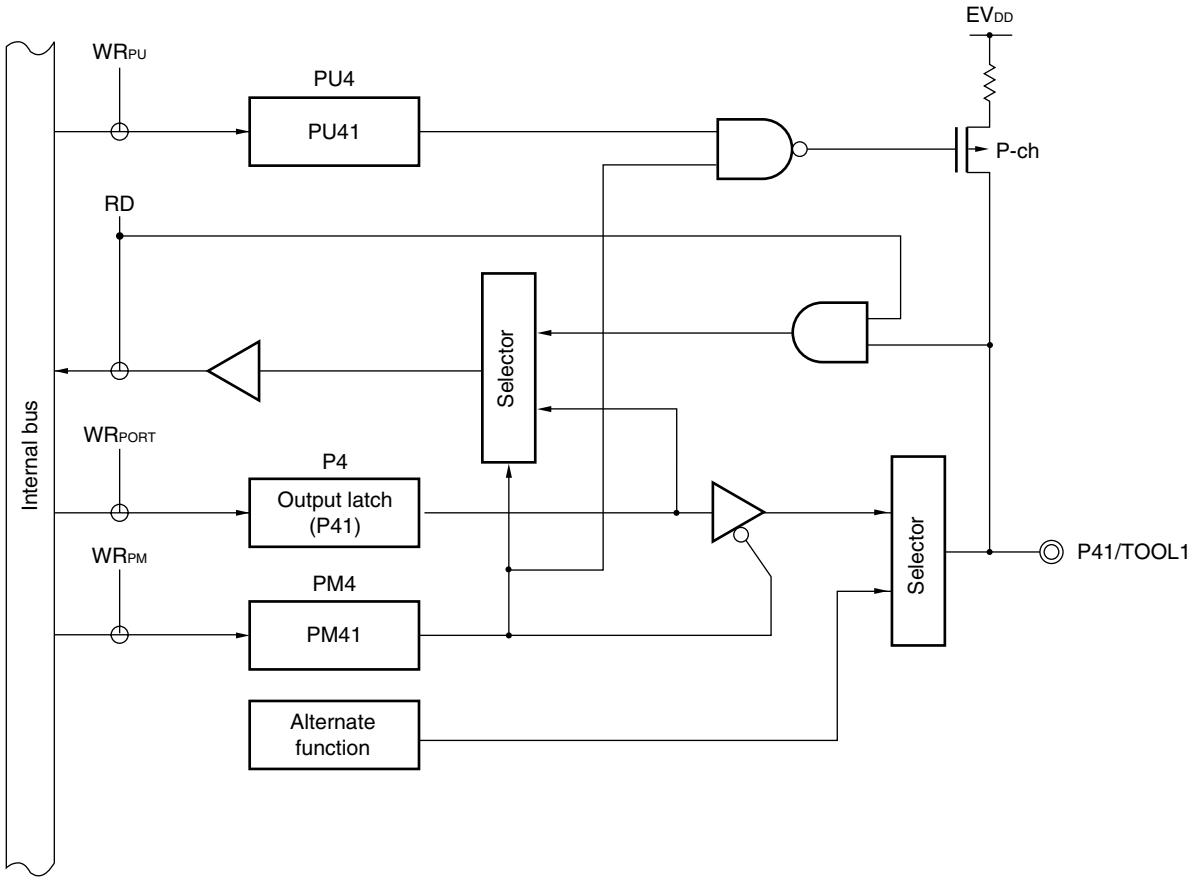
<R> **2. To use P42/TI04/TO04 as a general-purpose port, set bit 4 (TO04) of timer output register 0 (TO0) and bit 4 (TOE04) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.**

Figure 4-14. Block Diagram of P40



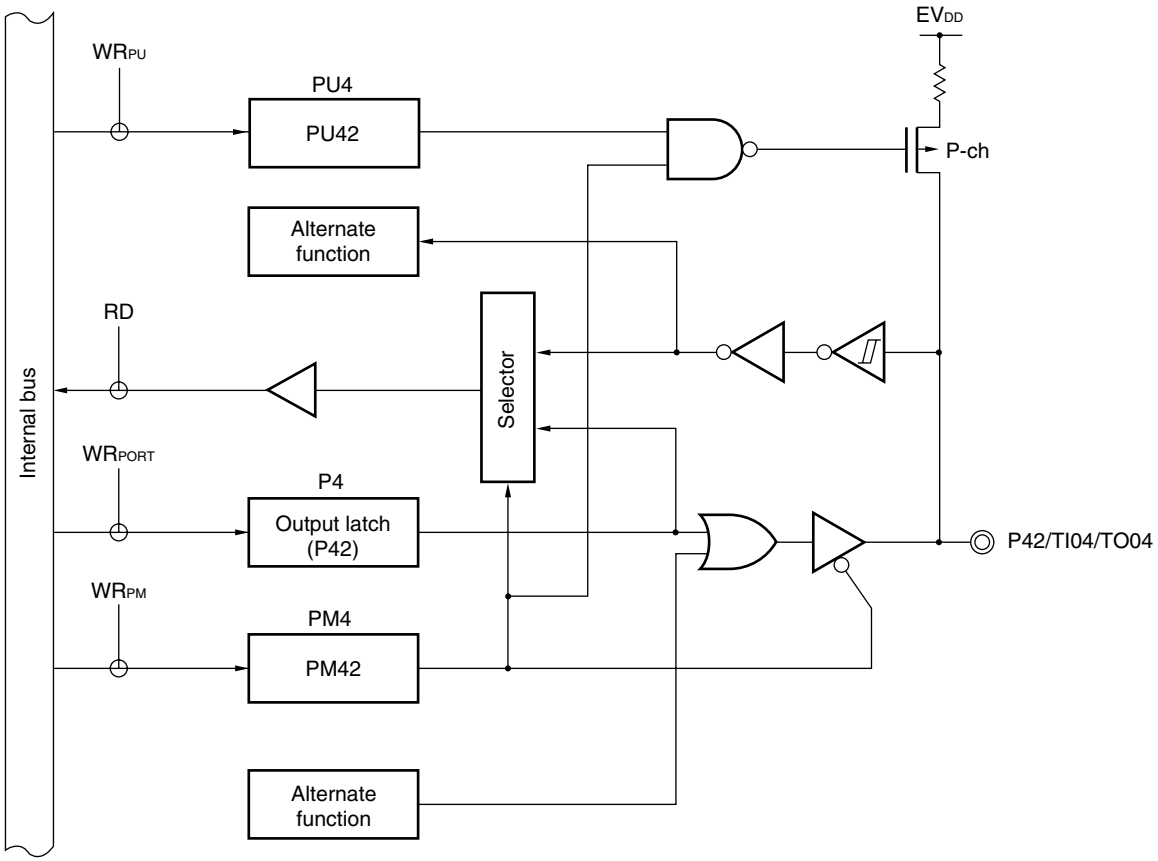
P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

Figure 4-15. Block Diagram of P41



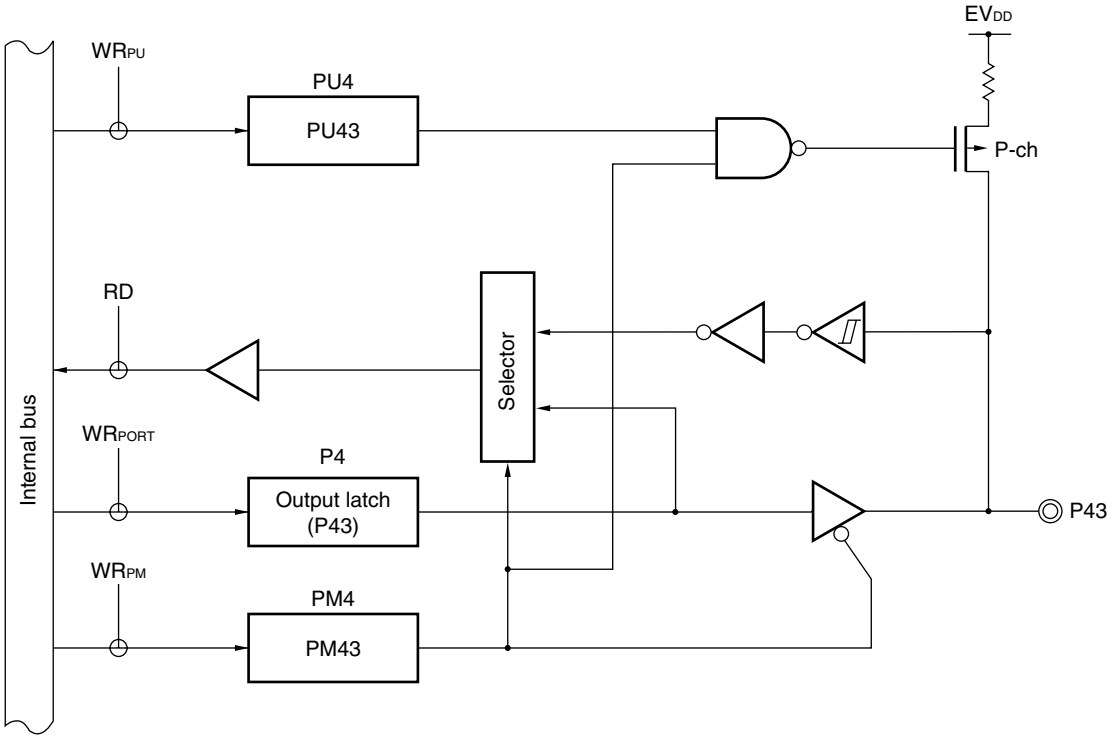
P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

Figure 4-16. Block Diagram of P42



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

Figure 4-17. 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

4.2.6 Port 5

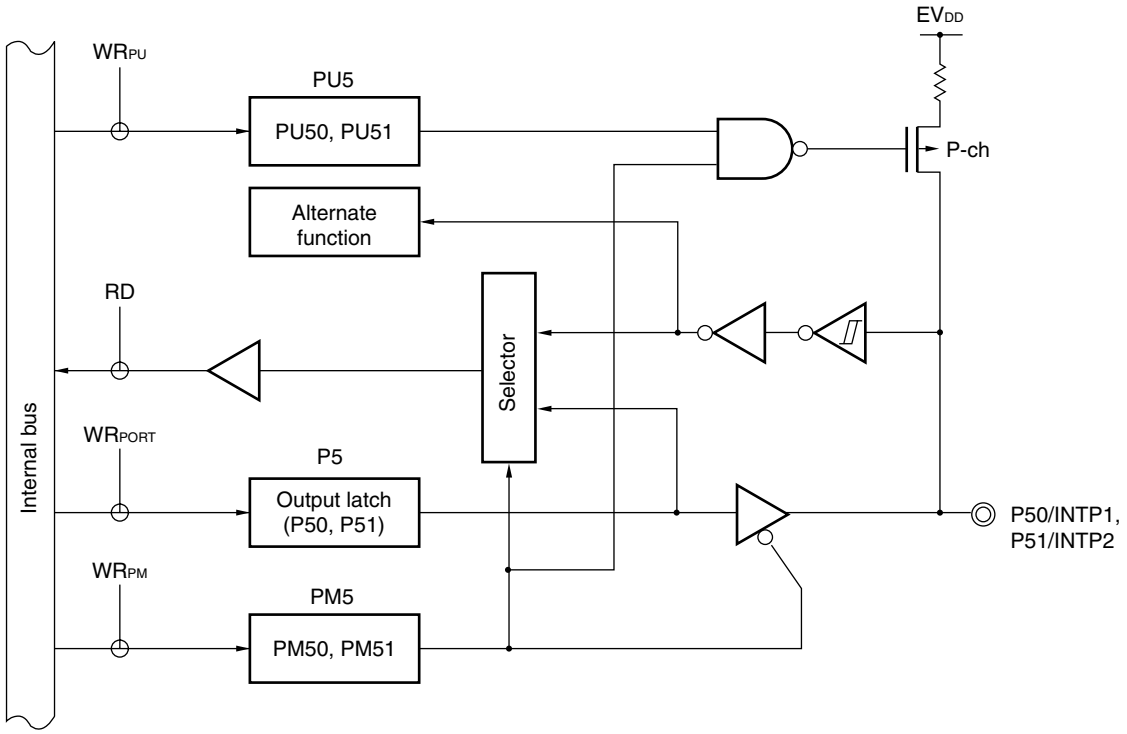
Port 5 is an 6-bit I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P55 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

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

Reset signal generation sets port 5 to input mode.

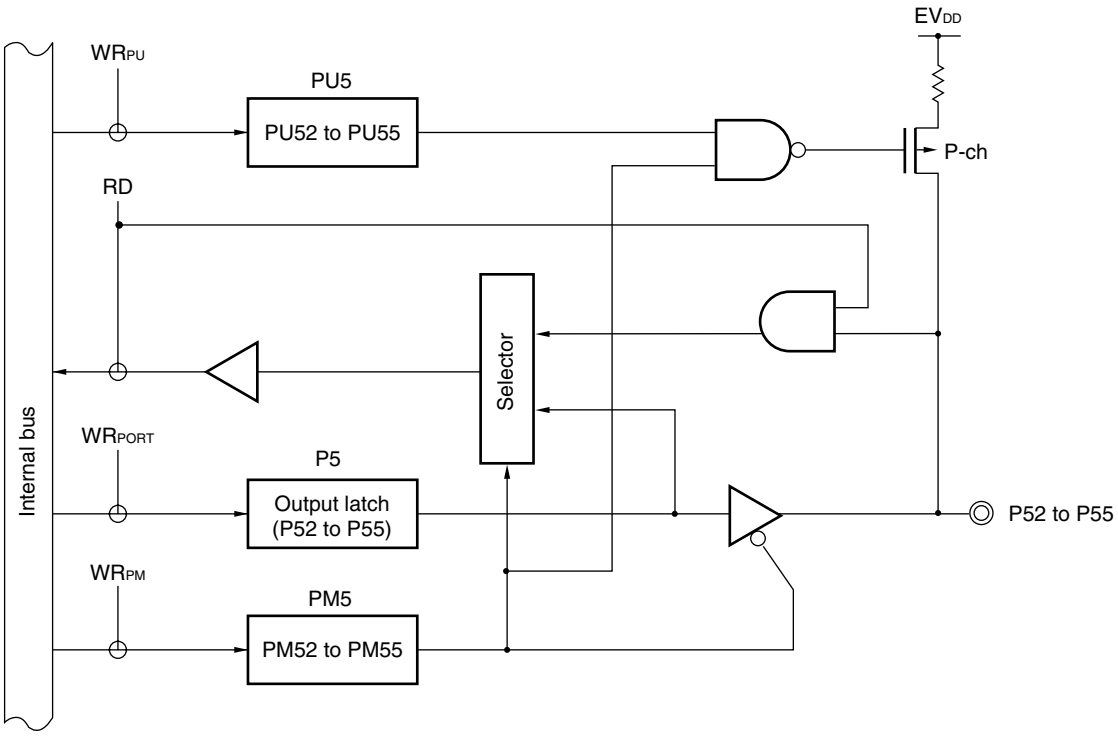
Figures 4-18 and 4-19 show block diagrams of port 5.

Figure 4-18. Block Diagram of P50 and P51



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

Figure 4-19. Block Diagram of P52 to P55



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 4-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).

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

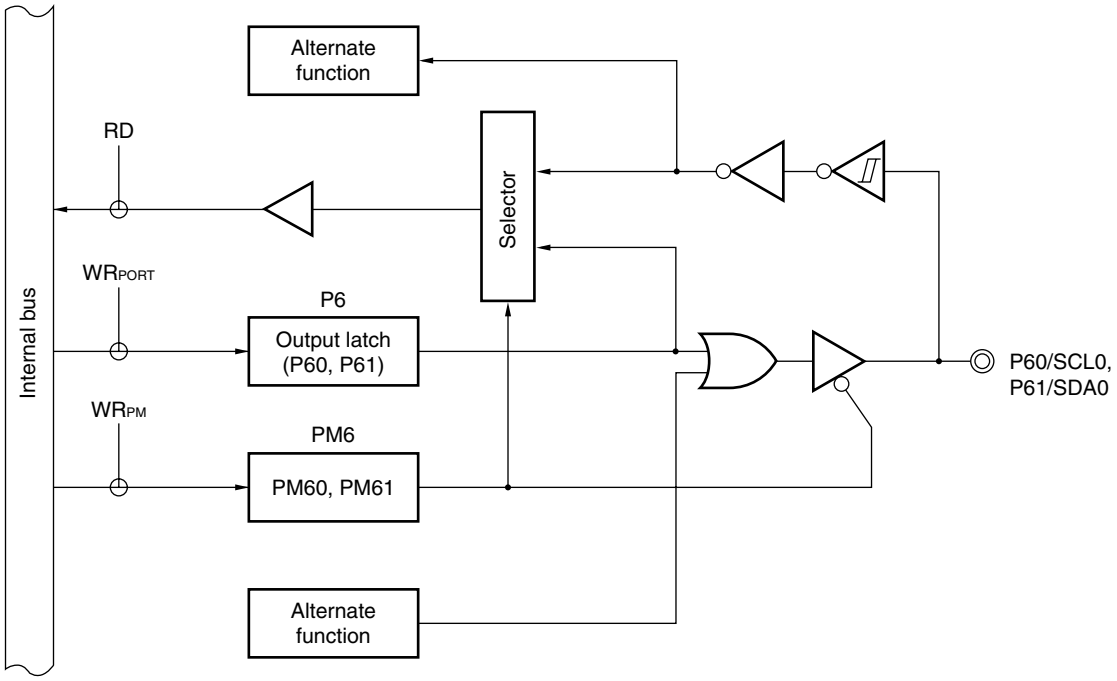
This port can also be used for serial interface data I/O and clock I/O.

Reset signal generation sets port 6 to input mode.

Figures 4-20 and 4-21 show block diagrams of port 6.

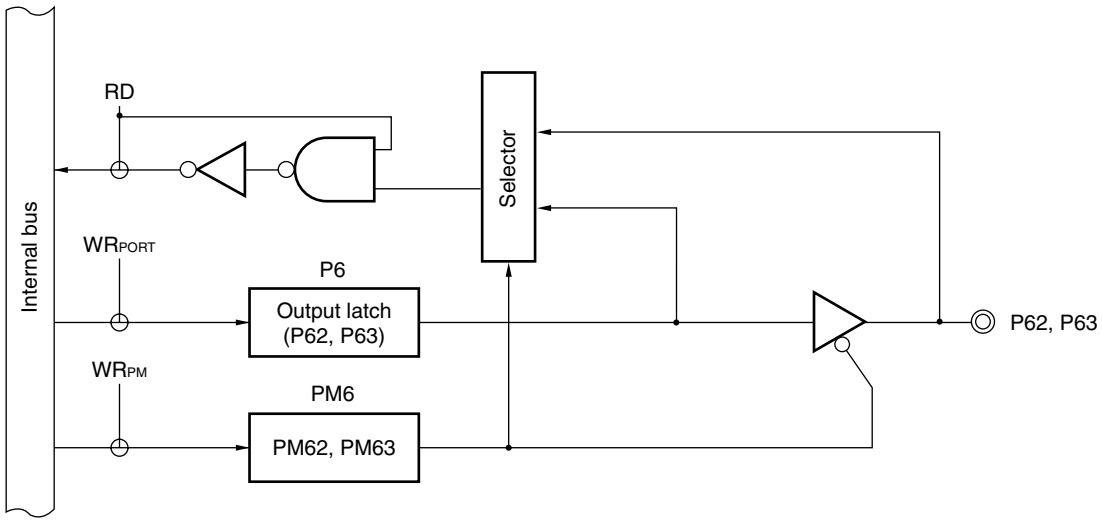
<R> **Caution When using P60/SCL0 or P61/SDA0 as a general-purpose port, stop the operation of serial interface IIC0.**

Figure 4-20. Block Diagram of P60 and P61



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

Figure 4-21. Block Diagram of P62 and P63



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

4.2.8 Port 7

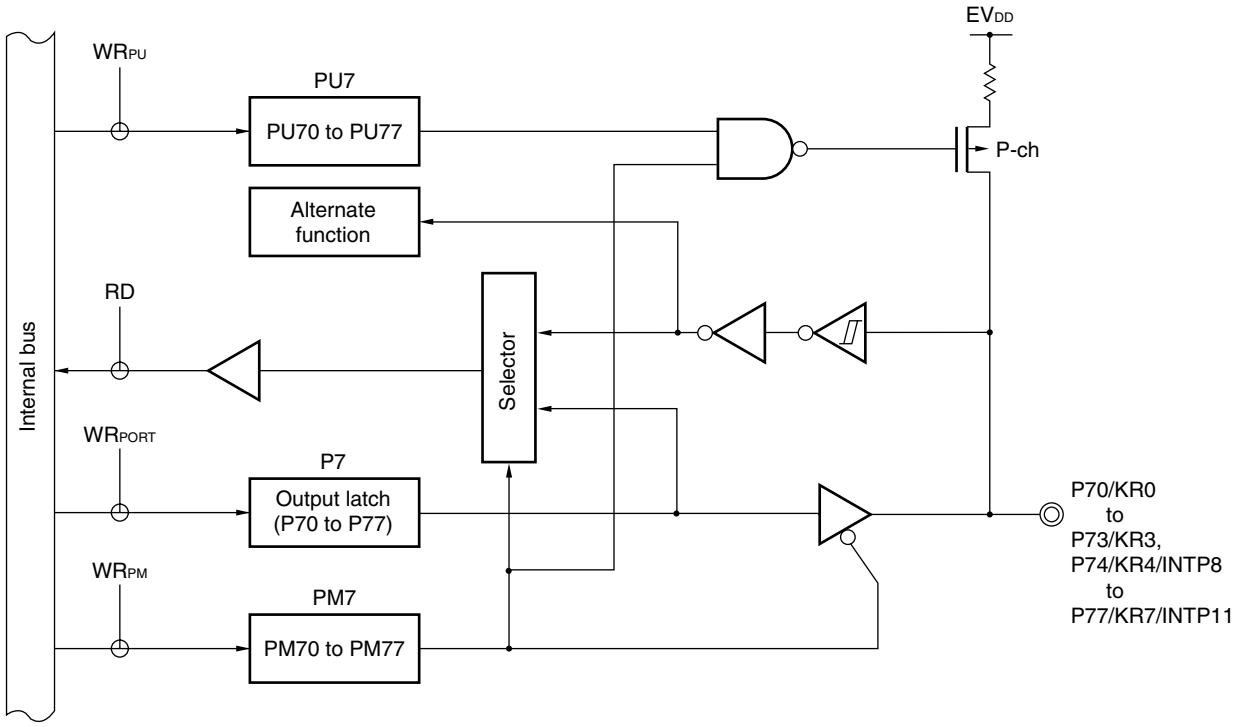
Port 7 is an 8-bit I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 to P77 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

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

Reset signal generation sets port 7 to input mode.

Figure 4-22 shows a block diagram of port 7.

Figure 4-22. 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

4.2.9 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.

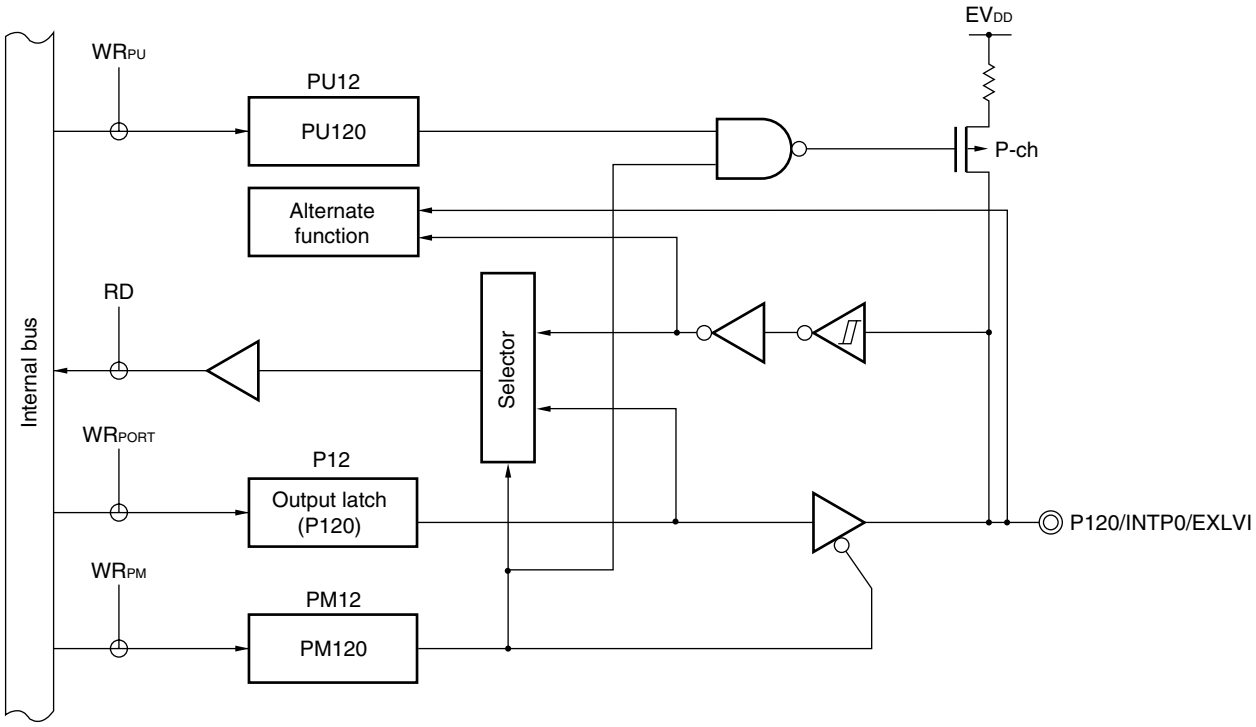
This port can also be used for external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, and external clock input for main system clock.

Reset signal generation sets port 12 to input mode.

Figures 4-23 to 4-25 show block diagrams of port 12.

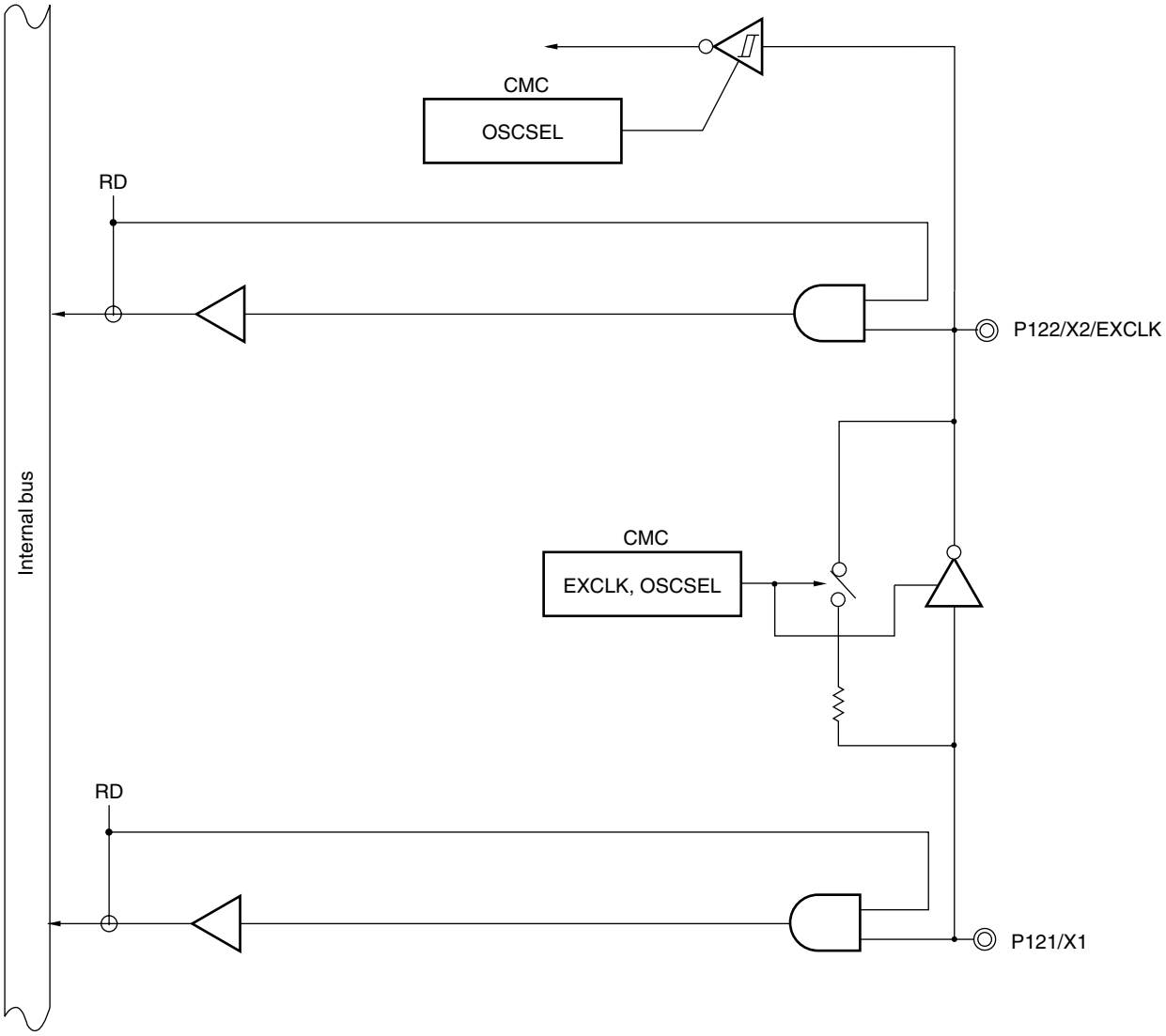
Caution The function setting on P121 to P124 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.

Figure 4-23. Block Diagram of P120



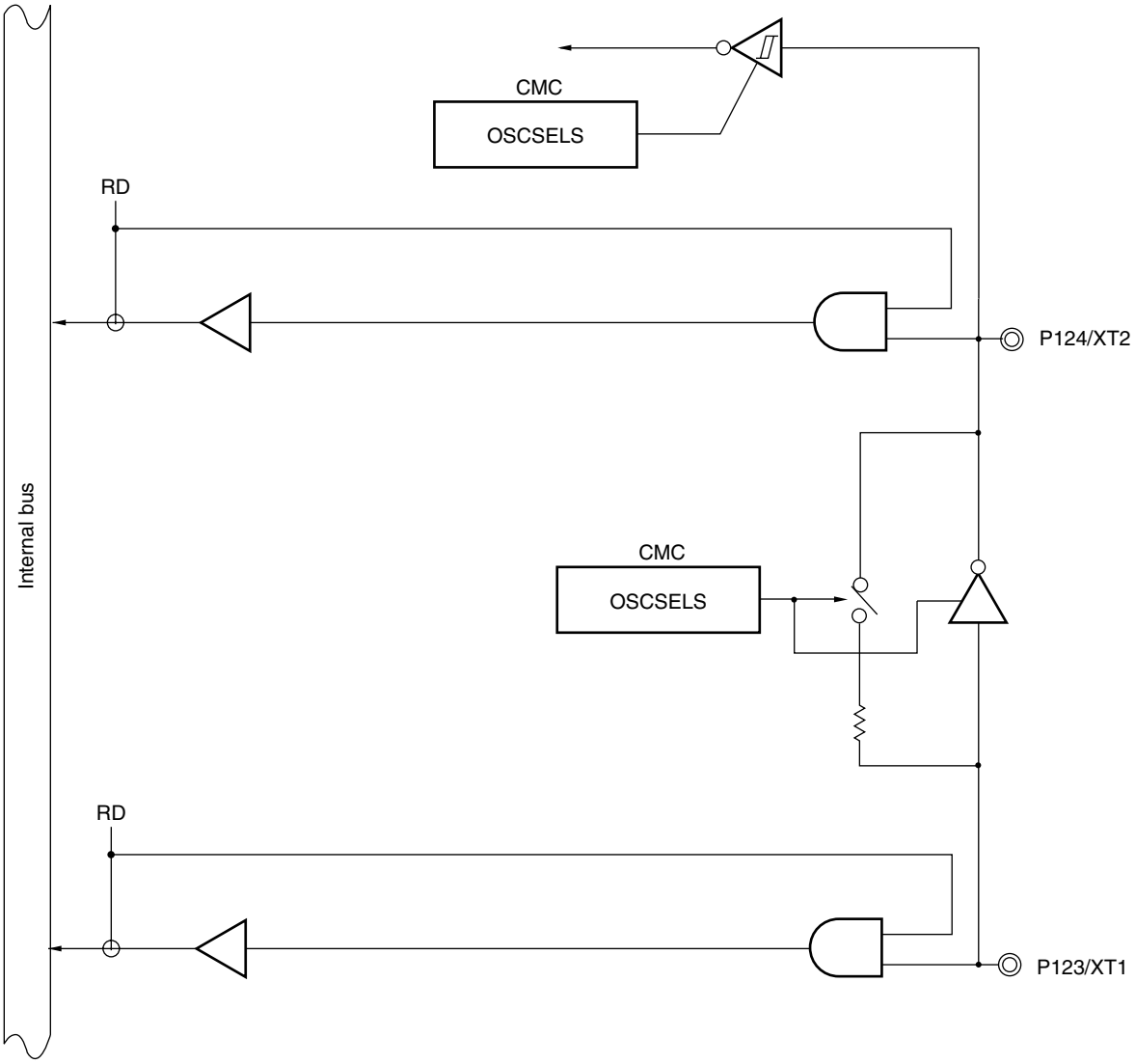
P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
RD: Read signal
WR××: Write signal

Figure 4-24. Block Diagram of P121 and P122



CMC: Clock operation mode control register
RD: Read signal

Figure 4-25. Block Diagram of P123 and P124



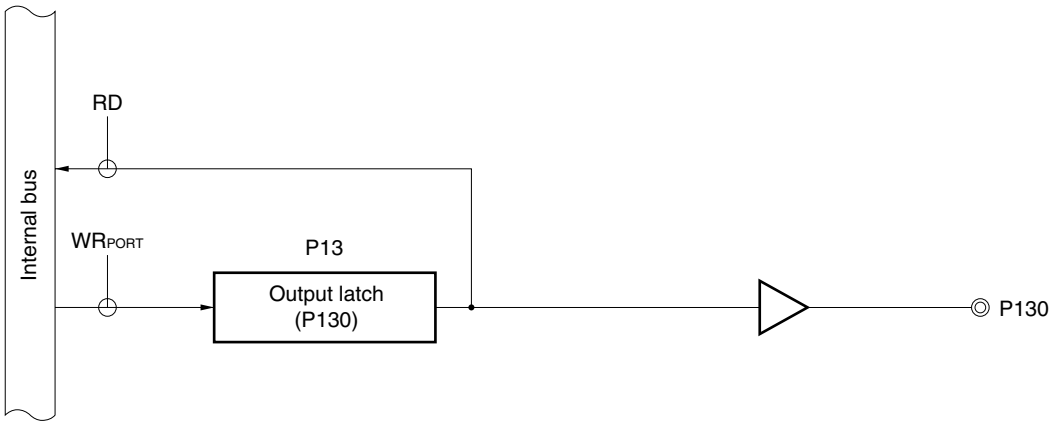
CMC: Clock operation mode control register
RD: Read signal

4.2.10 Port 13

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

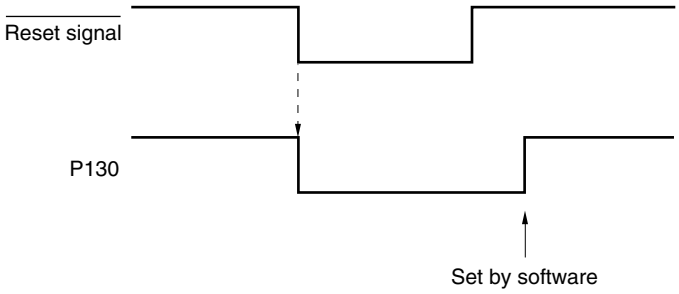
Figures 4-26 show block diagrams of port 13.

Figure 4-26. Block Diagram of P130



P13: Port register 13
RD: Read signal
WR××: Write signal

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



4.2.11 Port 14

Port 14 is a 2-bit I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140 and P141 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).

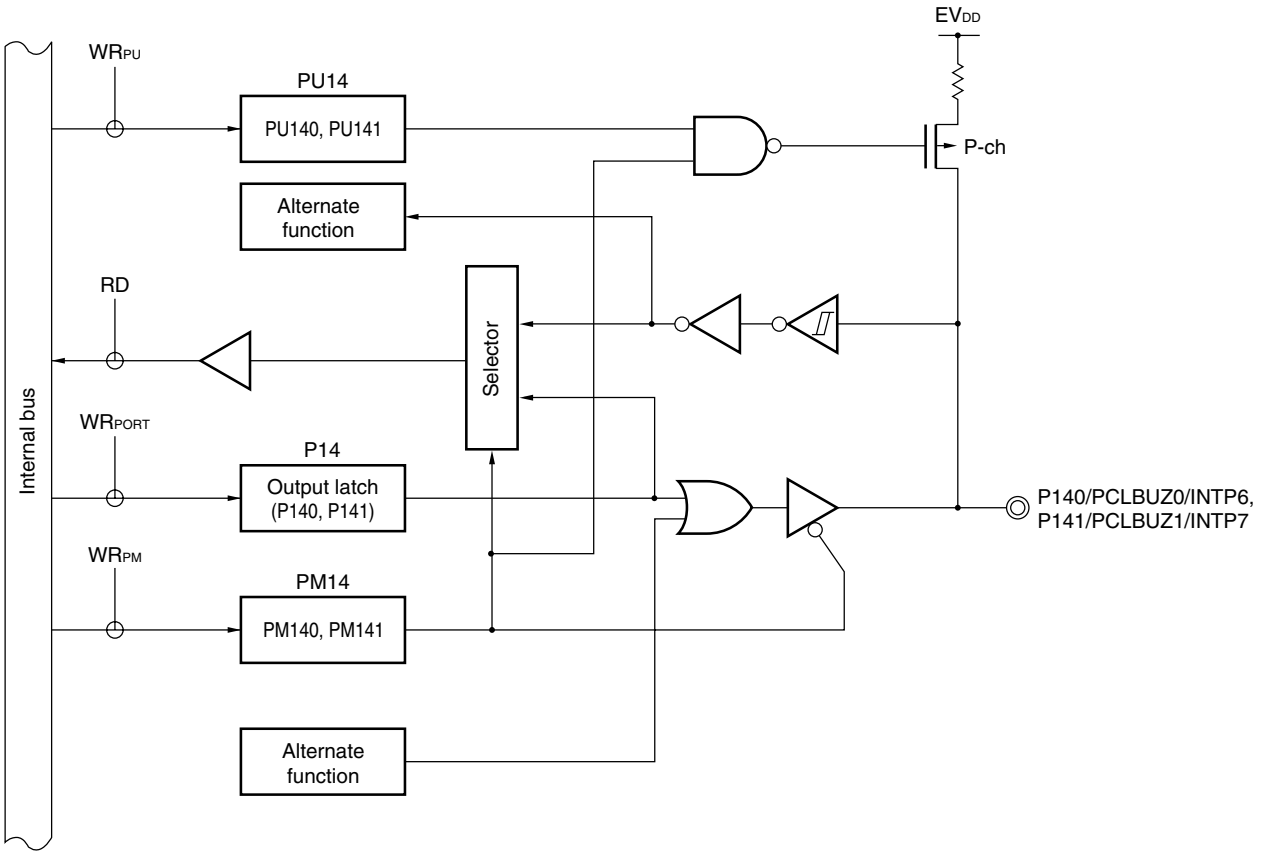
This port can also be used for external interrupt request input and clock/buzzer output.

Reset signal generation sets port 14 to input mode.

Figures 4-27 show block diagrams of port 14.

<R> **Caution To use P140/PCLBUZ0/INTP6 or P141/PCLBUZ1/INTP7 as a general-purpose port, set bit 7 of clock output select register 0 and 1 (CKS0, CKS1) to "0", which is the same as their default status setting.**

Figure 4-27. Block Diagram of P140 and P141



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

4.3 Registers Controlling Port Function

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

- Port mode registers (PM0 to PM7, PM12, PM14)
- Port registers (P0 to P7, P12 to P14)
- Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU7, PU12, PU14)
- Port input mode registers (PIM0)
- Port output mode registers (POM0)
- A/D port configuration register (ADPC)

(1) Port mode registers (PM0 to PM7, PM12, PM14)

These registers specify input or output mode for the port in 1-bit units.

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

Reset signal generation sets these registers to FFH.

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

Figure 4-28. Format of Port Mode Register

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	1	1	1	1	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	1	1	1	1	1	1	PM141	PM140	FFF2EH	FFH	R/W

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

Caution Be sure to set bit 7 of PM0, bits 2 to 7 of PM3, bits 4 to 7 of PM4, bits 6 and 7 of PM5, bits 4 to 7 of PM6, bits 1 to 7 of PM12, and bits 2 to 7 of PM14 to “1”.

(2) Port registers (P0 to P7, P12 to P14)

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 P2 is read during the input mode when P2 is set to function as an analog input for a A/D converter.

Figure 4-29. Format of Port Register

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	0	0	0	0	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	0	0	0	0	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
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	0	0	0	P141	P140	FFF0EH	00H (output latch)	R/W

Pmn	m = 0 to 7, 12 to 14; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note P121 to P124 are read-only.

(3) Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU7, PU12, PU14)

These registers specify whether the on-chip pull-up resistors of P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P140, or P141 are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode of the pins to which the use of an on-chip pull-up resistor has been specified in PU0, PU1, PU3 to PU5, PU7, PU12, and PU14. On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of PU0, PU1, PU3 to PU5, PU7, PU12, and PU14.

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

Reset signal generation clears these registers to 00H.

Figure 4-30. Format of Pull-up Resistor Option Register

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	0	0	0	0	PU43	PU42	PU41	PU40	F0034H	00H	R/W
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	0	0	0	0	0	0	PU141	PU140	F003EH	00H	R/W

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

(4) Port input mode registers (PIM0)

This register sets the input buffer of P03 or P04 in 1-bit units.

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

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

Reset signal generation clears this register to 00H.

Figure 4-31. Format of Port Input Mode Register

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

PIM0n	P0n pin input buffer selection (n = 3, 4)
0	Normal input buffer
1	TTL input buffer

(5) Port output mode registers (POM0)

This register sets the output mode of P02 to P04 in 1-bit units.

<R> 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.

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

Reset signal generation clears this register to 00H.

Figure 4-32. Format of Port Input Mode Register

Address: F0050H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
<R> POM0	0	0	0	POM04	POM03	POM02	0	0

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

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

This register switches the P20/ANI0 to P27/ANI7 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 4-33. 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							
					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
0	0	0	0	1	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	D	D	D	D	D	D
0	0	1	1	1	A	D	D	D	D	D	D	D
0	1	0	0	0	D	D	D	D	D	D	D	D
1	0	0	0	0	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 (PM2).

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

3. When using all ANI0/P20 to ANI7/P27 pins as digital I/O (D), the setting can be done by ADPC4 to ADPC0 = either 01000 or 10000.

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 power potential (2.5V, 3 V)

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

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

Moreover, regarding outputs, different power 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).

(1) Setting procedure when using I/O pins of UART1 and CSI10 functions

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

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

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

In case of UART1: P03
In case of CSI10: P03, P04

<3> Set the corresponding bit of the PIM0 register to 1 to switch to the TTL input buffer.

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

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

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

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

In case of UART1: P02
In case of CSI10: P02, P04

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

<4> Set the corresponding bit of the POM0 register to 1 to set the N-ch open drain output (V_{DD} withstand voltage) mode.

<5> Set the output mode by manipulating the PM0 register.
At this time, the output data is high level, so the pin is in the Hi-Z state.

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

(2) Setting procedure when using I/O pins of simplified IIC10 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

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

<4> Set the corresponding bit of the POM0 register to 1 to set the N-ch open drain output (V_{DD} withstand voltage) mode.

<5> Set the corresponding bit of the PM0 register to the output mode (data I/O is possible in the output mode).

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

<6> Enable the operation of the serial array unit and set the mode to the simplified I^2C mode.

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

To use the alternate function of a port pin, set the port mode register and output latch as shown in Table 4-5.

Table 4-5. Settings of Port Mode Register and Output Latch When Using Alternate Function (1/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P00	TI00	Input	1	×
P01	TO00	Output	0	0
P02	SO10	Output	0	1
	TxD1	Output	0	1
P03	SI10	Input	1	×
	RxD1	Input	1	×
	SDA10	I/O	0	1
P04	$\overline{\text{SCK10}}$	Input	1	×
		Output	0	1
	SCL10	I/O	0	1
P05	TI05	Input	1	×
	TO05	Output	0	0
P06	TI06	Input	1	×
	TO06	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
P13	TxD3	Output	0	1
P14	RxD3	Input	1	×
P15	RTCDIV	Output	0	0
	RTCCL	Output	0	0
P16	TI01	Input	1	×
	TO01	Output	0	0
	INTP5	Input	1	×

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

Table 4-5. Settings of Port Mode Register and Output Latch When Using Alternate Function (2/2)

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P17	TI02	Input	1	×
	TO02	Output	0	0
P20 to P27[Note]	ANI0 to ANI7[Note]	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
P50	INTP1	Input	1	×
P51	INTP2	Input	1	×
P60	SCL0	I/O	0	0
P61	SDA0	I/O	0	0
P70 to P73	KR0 to KR3	Input	1	×
P74 to P77	INTP8 to INTP11	Input	1	×
	KR4 to KR7	Input	1	×
P120	INTP0	Input	1	×
	EXLVI	Input	1	×
P140	PCLBUZ0	Output	0	0
	INTP6	Input	1	×
P141	PCLBUZ1	Output	0	0
	INTP7	Input	1	×

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

Note The function of the ANI0/P20 to ANI7/P27 pins can be selected by using the A/D port configuration register (ADPC), the analog input channel specification register (ADS), and PM2.

Table 4-6. Setting Functions of ANI0/P20 to ANI7/P27 Pins

ADPC	PM2	ADS	ANI0/P20 to ANI7/P27 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.	

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/KE3.

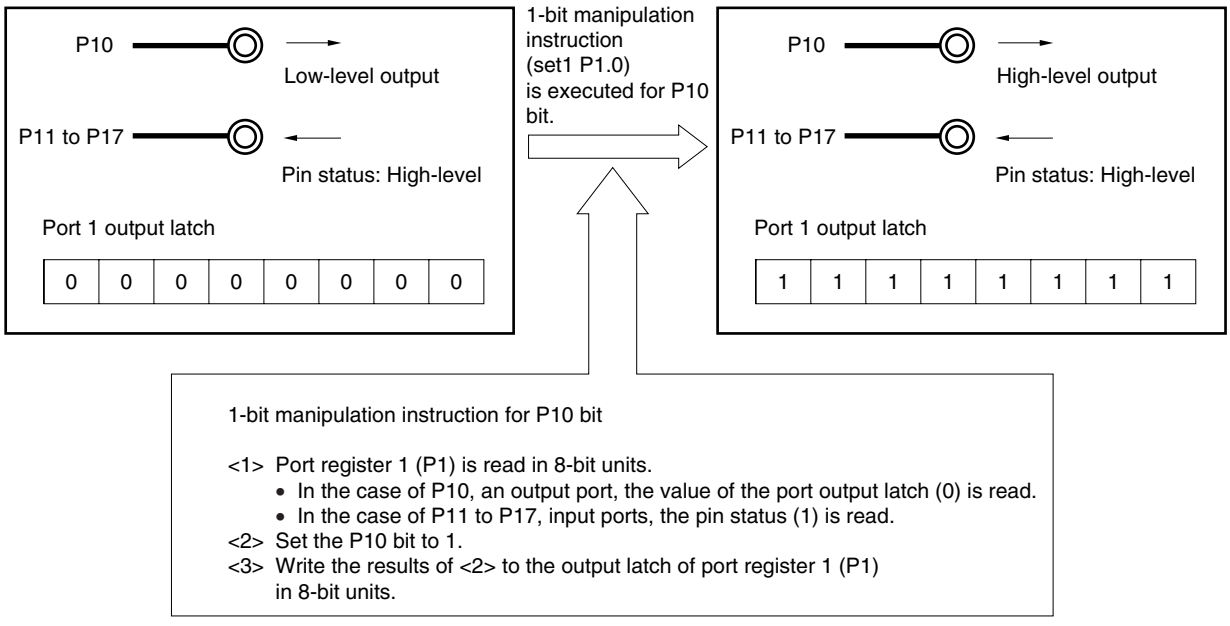
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

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 4-34. 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

This circuit oscillates a clock of f_{IH} = 8 MHz (TYP.). After a reset release, the CPU always starts operating with this internal high-speed oscillation clock. Oscillation can be stopped by executing the STOP instruction or setting of HIOSTOP (bit 0 of CSC).

An external main system clock (f_{EX} = 2 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of MSTOP.
As the main system clock, a high-speed system clock (X1 clock or external main system clock) or internal high-speed oscillation clock can be selected by setting of MCM0 (bit 4 of the system clock control register (CKC)).

(2) Subsystem clock

- **XT1 clock oscillator**

This circuit oscillates a clock of f_{SUB} = 32.768 kHz by connecting a 32.768 kHz resonator to XT1 and XT2.
Oscillation can be stopped by setting XTSTOP (bit 6 of CSC).

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

(3) Internal low-speed oscillation clock (clock for watchdog timer)

- **Internal low-speed oscillator**

This circuit oscillates a clock of f_{IL} = 240 kHz (TYP.).
The internal low-speed oscillation clock cannot be used as the CPU clock. The only hardware that operates with the internal low-speed oscillation clock is the watchdog timer.
Oscillation is stopped when the watchdog timer stops.

Remarks 1. f_{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

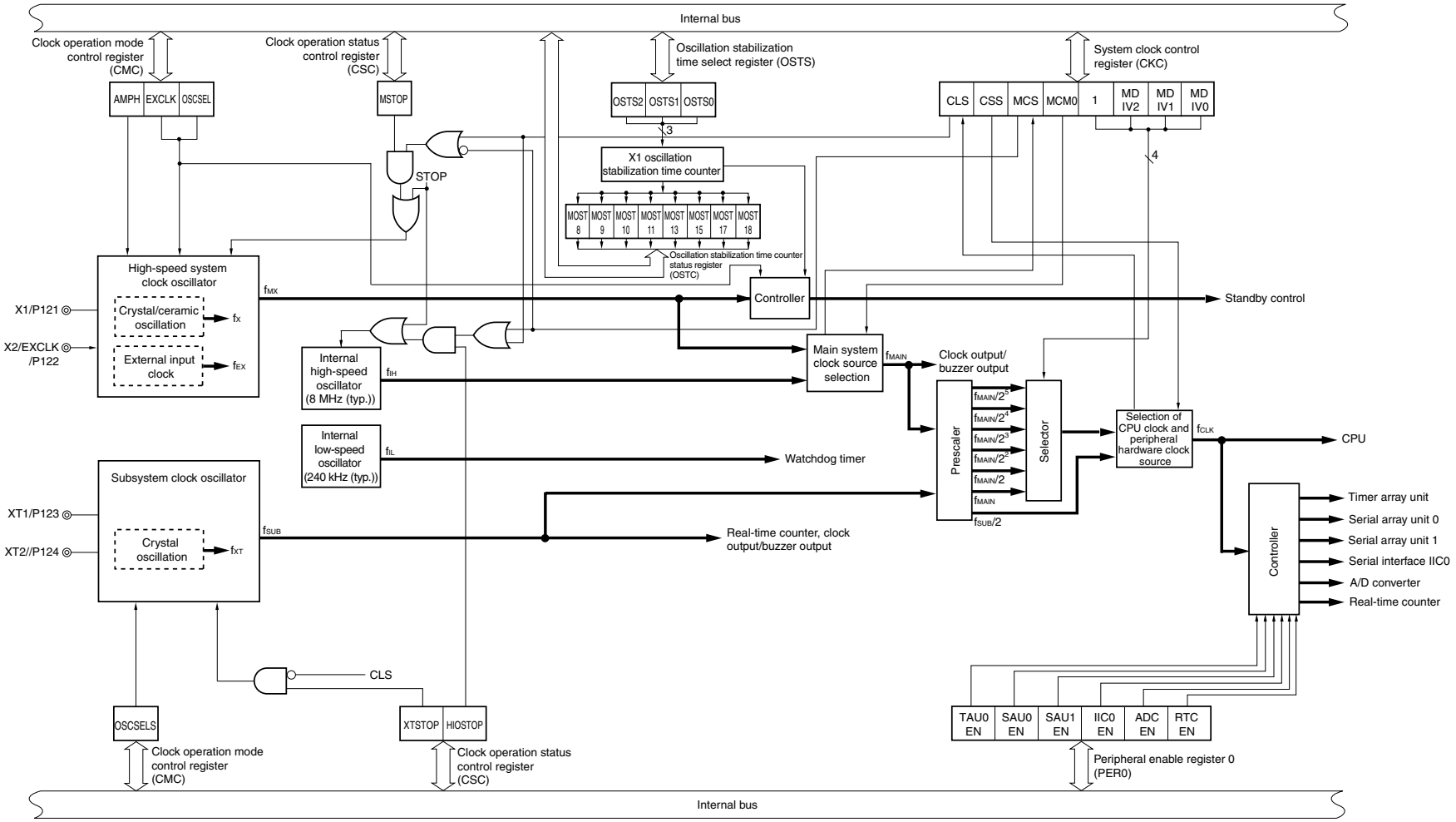
5.2 Configuration of Clock Generator

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) Peripheral enable register 0 (PER0) Operation speed mode control register (OSMC) Internal high-speed oscillator trimming register (HIOTRM)
Oscillators	X1 oscillator XT1 oscillator Internal high-speed oscillator Internal low-speed oscillator

Figure 5-1. Block Diagram of Clock Generator



Remark f_X: X1 clock oscillation frequency
f_{IH}: 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_{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 nine registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- System clock control register (CKC)
- Peripheral enable registers 0 (PER0)
- Operation speed mode control register (OSMC)
- Internal high-speed oscillator trimming register (HIOTRM)

(1) Clock operation mode control register (CMC)

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

CMC can be written only once by an 8-bit memory manipulation instruction after reset release. This register can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 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	0	0	AMPH

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

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

AMPH	Control of high-speed system clock oscillation frequency
0	2 MHz ≤ f_{MX} ≤ 10 MHz
1	10 MHz < f_{MX} ≤ 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. It is recommended to set the default value (00H) to CMC after reset release, even when the register is used at the default value, in order to prevent malfunctioning during a program loop.

Remark f_{MX}: High-speed system clock frequency

(2) Clock operation status control register (CSC)

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

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

Reset signal generation sets this register to C0H.

Figure 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	–
1	X1 oscillator stopped	External clock from EXCLK pin is invalid	

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

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

Cautions 1. After reset release, set the clock operation mode control register (CMC) before starting X1 oscillation as set by MSTOP or XT1 oscillation as set by XTSTOP.

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

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

Cautions 4. The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) and the condition before clock oscillation is to be stopped are as follows.

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	• CLS = 0 and MCS = 0 • CLS = 1 (CPU and peripheral hardware clocks operate with a clock other than the high-speed system clock.)	MSTOP = 1
Subsystem clock	• CLS = 0 (CPU and peripheral hardware clocks operate with a clock other than the subsystem clock.)	XTSTOP = 1
Internal high-speed oscillation clock	• CLS = 0 and MCS = 1 • CLS = 1 (CPU and peripheral hardware clocks operate with a clock other than the internal high-speed oscillator clock.)	HIOSTOP = 1

(3) Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.
The X1 clock oscillation stabilization time can be checked in the following case,

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

OSTC can be read by a 1-bit or 8-bit memory manipulation instruction.
When reset signal is generated, the STOP instruction and MSTOP (bit 7 of CSC register) = 1 clear OSTC to 00H.

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

Figure 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.

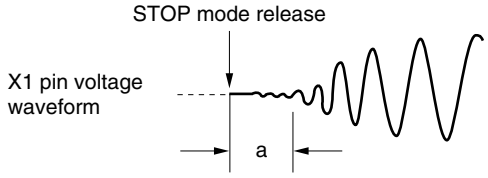
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.

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 the count value which is to be checked by the OSTC register after the oscillation starts.

- **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. The oscillation stabilization time can be checked up to the time set using OSTC.

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

Reset signal generation sets OSTS to 07H.

Figure 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. To change the setting of the OSTS register, be sure to confirm that the counting operation of the OSTC register has been completed.

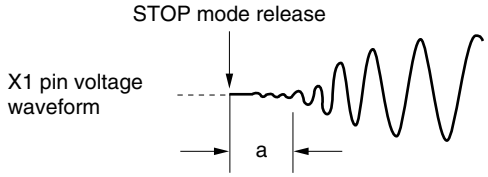
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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

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

- **If the X1 clock starts oscillation while the 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 f_X: X1 clock oscillation frequency

(5) System clock control register (CKC)

This register is used to select a CPU/peripheral hardware clock and a division ratio.

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

Reset signal generation sets this register to 09H.

Figure 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 (f_{SUB})

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

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

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

2. Setting is prohibited when f_{MX} < 4 MHz.

<R> **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_{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 operating hardware clock.

3. If the peripheral hardware clock is used as the subsystem clock, the operations of the A/D converter and IIC0 are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 27 ELECTRICAL SPECIFICATIONS.

The fastest instruction can be executed in 1 clock of the CPU clock in the 78K0R/KE3. 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 32.768 kHz Operation
f_{MAIN}	0.1 μs	0.05 μs	0.125 μs (TYP.)	–
$f_{MAIN}/2$	0.2 μs	0.1 μs	0.25 μs (TYP.) (default)	–
$f_{MAIN}/2^2$	0.4 μs	0.2 μs	0.5 μs (TYP.)	–
$f_{MAIN}/2^3$	0.8 μs	0.4 μs	1.0 μs (TYP.)	–
$f_{MAIN}/2^4$	1.6 μs	0.8 μs	2.0 μs (TYP.)	–
$f_{MAIN}/2^5$	3.2 μs	1.6 μs	4.0 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) Peripheral enable registers 0 (PER0)

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

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

Reset signal generation clears theses registers to 00H.

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

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCEN	0	ADCEN	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN

RTCEN	Control of real-time counter (RTC) input clock[Note]
0	Stops input clock supply. • SFR used by the real-time counter (RTC) cannot be written (can be read). • Operation of the real-time counter (RTC) continues.
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
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.

IIC0EN	Control of serial interface IIC0 input clock
0	Stops input clock supply. • SFR used by the serial interface IIC0 cannot be written. • The serial interface IIC0 is in the reset status.
1	Supplies input clock. • SFR used by the serial interface IIC0 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 bits 1 and 6 of PER0 register to 0.

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

SAU1EN	Control of serial array unit 1 input clock
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
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.

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

Caution Be sure to clear bits 1 and 6 of PER0 register to 0.

(7) Operation speed mode control register (OSMC)

This register is used to control the step-up circuit of the flash memory for high-speed operation.

If the microcontroller operates at a low speed with a system clock of 10 MHz or less, the power consumption can be lowered by setting this register to the default value, 00H.

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

Reset signal generation clears this register to 00H.

Figure 5-8. Format of Operation Speed Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	0	0	0	0	0	0	0	FSEL

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

Cautions 1. OSMC can be written only once after reset release, by an 8-bit memory manipulation instruction.

2. Write "1" to FSEL before the following two operations.

- **Changing the clock prior to dividing f_{CLK} to a clock other than f_{IH}.**
- **Operating the DMA controller.**

3. The CPU waits when "1" is written to the FSEL flag.
The wait time is 15 μs to 20 μs (target) when $f_{CLK} = f_{IH}$, and 30 μs to 40 μs (target) when $f_{CLK} = f_{IH}/2$.
However, counting the oscillation stabilization time of f_X can continue even while the CPU is waiting.

4. To increase f_{CLK} to 10 MHz or higher, set FSE L to "1", then change f_{CLK} after two or more clocks have elapsed.

<R> **5. Even when set to FSEL = 1, the system clock can be operated at a frequency of 10 MHz or less.**
When setting FSEL to "1", however, do so while $V_{DD} \geq 2.25$ V.
When set to FSEL = 1, make sure that $V_{DD} \geq 2.25$ V at the following timings, even if f_{CLK} is divided.

- **When releasing f_{IH} or f_{EX} from the STOP mode selected for f_{CLK}**
- **When switching f_{CLK} from f_{SUB} to f_{MAIN}**

(8) Internal high-speed oscillator trimming register (HIOTRM)

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

With self-measurement of the internal high-speed oscillator frequency via a subsystem clock using a crystal resonator, a timer using high-accuracy external clock input (real-time counter or timer array unit), and so on, the register can adjust the accuracy.

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

Reset signal generation sets this register to 10H.

Caution The frequency will vary if the temperature and V_{DD} pin voltage change after accuracy adjustment. Moreover, if the HIOTRM register is set to any value other than the initial value (10H), the oscillation accuracy of the internal high-speed oscillation clock may exceed 8 MHz±5%, depending on the subsequent temperature and V_{DD} voltage change, or HIOTRM register setting. When the temperature and V_{DD} voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.

Figure 5-9. Format of Internal High-Speed Oscillator Trimming Register (HIOTRM)

Address: F00F2H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
HIOTRM	0	0	0	TTRM4	TTRM3	TTRM2	TTRM1	TTRM0

TTRM4	TTRM3	TTRM2	TTRM1	TTRM0	Clock correction value (target) (2.7 V ≤ V_{DD} ≤ 5.5 V)		
					MIN.	TYP.	MAX.
0	0	0	0	0	–5.54%	–4.88%	–4.02%
0	0	0	0	1	–5.28%	–4.62%	–3.76%
0	0	0	1	0	–4.99%	–4.33%	–3.47%
0	0	0	1	1	–4.69%	–4.03%	–3.17%
0	0	1	0	0	–4.39%	–3.73%	–2.87%
0	0	1	0	1	–4.09%	–3.43%	–2.57%
0	0	1	1	0	–3.79%	–3.13%	–2.27%
0	0	1	1	1	–3.49%	–2.83%	–1.97%
0	1	0	0	0	–3.19%	–2.53%	–1.67%
0	1	0	0	1	–2.88%	–2.22%	–1.36%
0	1	0	1	0	–2.23%	–1.91%	–1.31%
0	1	0	1	1	–1.92%	–1.60%	–1.28%
0	1	1	0	0	–1.60%	–1.28%	–0.96.%
0	1	1	0	1	–1.28%	–0.96%	–0.64%
0	1	1	1	0	–0.96%	–0.64%	–0.32%
0	1	1	1	1	–0.64%	–0.32%	±0%
1	0	0	0	0	±0% (default)		
1	0	0	0	1	+0%	+0.32%	+0.64%
1	0	0	1	0	+0.33%	+0.65%	+0.97%
1	0	0	1	1	+0.66%	+0.98%	+1.30%
1	0	1	0	0	+0.99%	+1.31%	+1.63%
1	0	1	0	1	+1.32%	+1.64%	+1.96%
1	0	1	1	0	+1.38%	+1.98%	+2.30%
1	0	1	1	1	+1.46%	+2.32%	+2.98%
1	1	0	0	0	+1.80%	+2.66%	+3.32%
1	1	0	0	1	+2.14%	+3.00%	+3.66%
1	1	0	1	0	+2.48%	+3.34%	+4.00%
1	1	0	1	1	+2.83%	+3.69%	+4.35%
1	1	1	0	0	+3.18%	+4.04%	+4.70%
1	1	1	0	1	+3.53%	+4.39%	+5.05%
1	1	1	1	0	+3.88%	+4.74%	+5.40%
1	1	1	1	1	+4.24%	+5.10%	+5.76%

Caution The internal high-speed oscillation frequency becomes faster/slower by increasing/decreasing the HIOTRM value to a value larger/smaller than a certain value. A reversal, such as the frequency becoming slower/faster by increasing/decreasing the HIOTRM value does not occur.

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

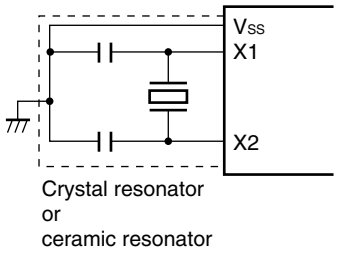
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

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

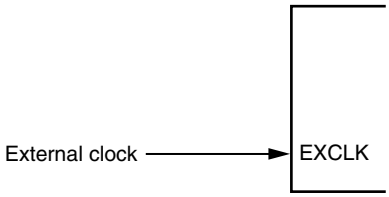
Figure 5-10 shows an example of the external circuit of the X1 oscillator.

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.

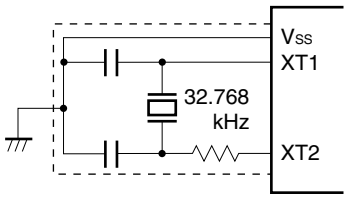
To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

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 2-2 Connection of Unused Pins**.

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 1. 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.

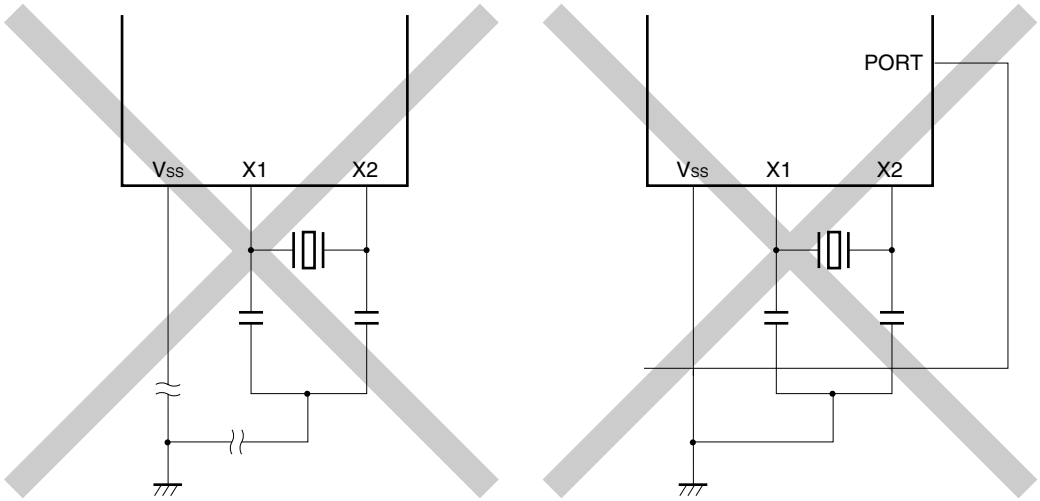
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground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

Note that the XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption.

Figure 5-12 shows examples of incorrect resonator connection.

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

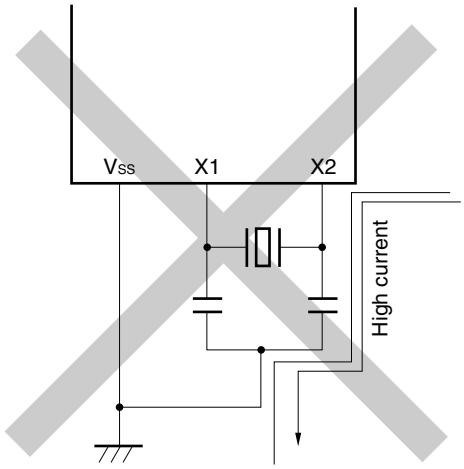
(a) Too long wiring **(b) Crossed signal line**



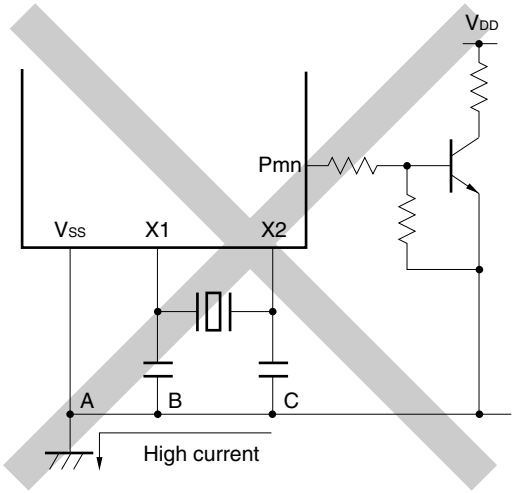
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

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

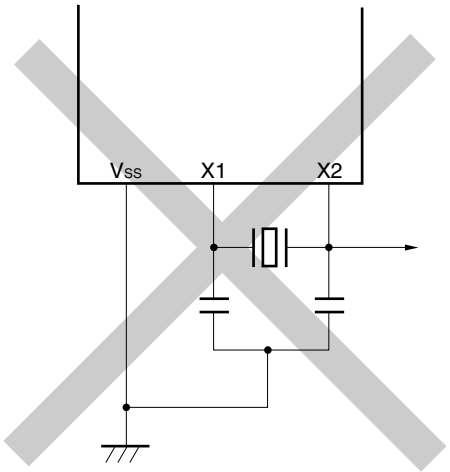
(c) Wiring near high alternating current



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



(e) Signals are fetched



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

Caution 2. When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.

5.4.3 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0R/KE3 (8 MHz (TYP.)). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC).

After a reset release, the internal high-speed oscillator automatically starts oscillation.

5.4.4 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0R/KE3.

The internal low-speed oscillation clock is used only as the watchdog timer clock. The internal low-speed oscillation clock cannot be used as the CPU clock.

After a reset release, the internal low-speed oscillator automatically starts oscillation, and the watchdog timer is driven (240 kHz (TYP.)) if the watchdog timer operation is enabled by the option byte.

The internal low-speed oscillator continues oscillation except when the watchdog timer stops. When the watchdog timer operates, the internal low-speed oscillation clock does not stop, even in case of a program loop.

5.4.5 Prescaler

The prescaler generates 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}
- 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/KE3, thus enabling the following.

(1) Enhancement of security function

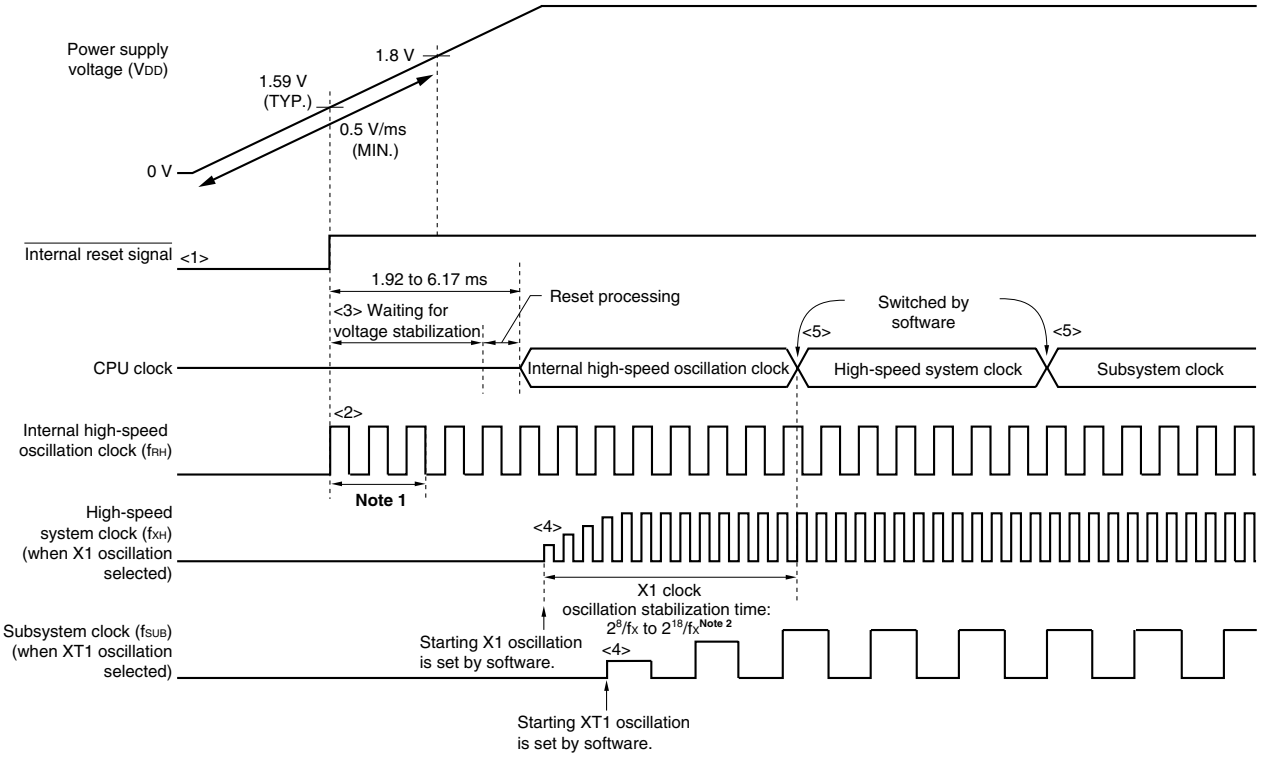
When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the internal high-speed oscillation clock, so the device can be started by the internal high-speed oscillation clock after a reset release. As a result, reset sources can be detected by software and the minimum amount of safety processing can be done during anomalies to ensure that the system terminates safely.

(2) Improvement of performance

Because the CPU can be started without waiting for the X1 clock oscillation stabilization time, the total performance can be improved.

When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-13 and Figure 5-14.

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

<3> When the power supply voltage rises with a slope of 0.5 V/ms (MIN.), the CPU starts operation on the internal high-speed oscillation clock after the reset is released and after the stabilization times for the voltage of the power supply and regulator have elapsed, and then reset processing is performed.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **(1)** in **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 **(3)** in **5.6.3 Example of controlling subsystem clock**).

Notes 1. The internal voltage stabilization time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

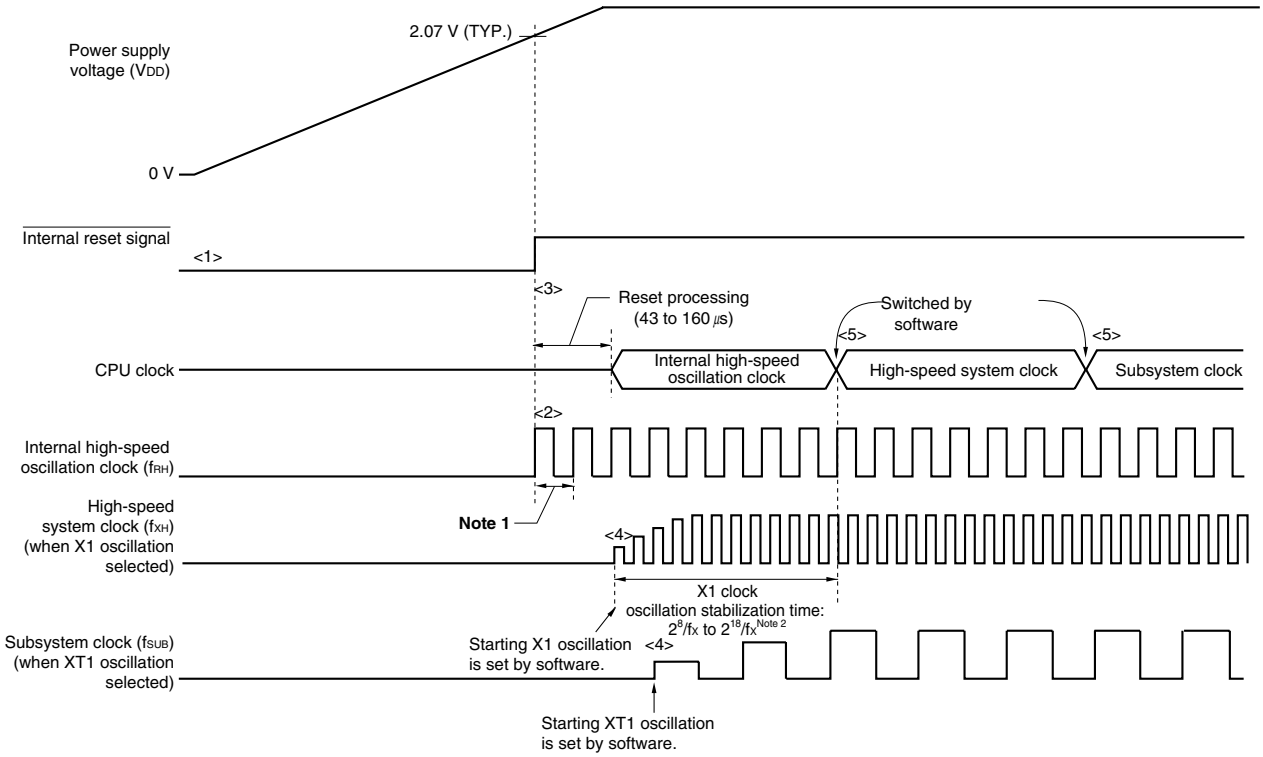
2. When releasing a reset (above figure) or releasing STOP mode while the CPU is operating on the internal high-speed oscillation clock, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC). If the CPU operates on the high-speed system clock (X1 oscillation), set the oscillation stabilization time when releasing STOP mode using the oscillation stabilization time select register (OSTS).

Cautions 1. If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 1.8 V, input a low level to the $\overline{\text{RESET}}$ pin from power application until the voltage reaches 1.8 V, or set the LVI default start function stopped by using the option byte (LVIOFF = 0) (see Figure 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.

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

Remark While the microcontroller is operating, a clock that is not used as the CPU clock can be stopped via software settings. The internal high-speed oscillation clock and high-speed system clock can be stopped by executing the STOP instruction (see **(4)** in **5.6.1 Example of controlling high-speed system clock**, **(3)** in **5.6.2 Example of controlling internal high-speed oscillation clock**, and **(4)** 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 power-on-clear (POC) 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 **(3)** 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 oscillation stabilization time is required after the supply voltage reaches 1.59 V (TYP.). If the supply voltage rises from 1.59 V (TYP.) to 2.07 V (TYP.) within the power supply oscillation stabilization time, the power supply oscillation stabilization time is automatically generated before reset processing.

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

Remark While the microcontroller is operating, a clock that is not used as the CPU clock can be stopped via software settings. The internal high-speed oscillation clock and high-speed system clock can be stopped by executing the STOP instruction (see **(4)** in **5.6.1 Example of controlling high-speed system clock**, **(3)** in **5.6.2 Example of controlling internal high-speed oscillation clock**, and **(4)** in **5.6.3 Example of controlling subsystem clock**).

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	0	0	AMPH
0	1	0	0/1	0	0	0	0

- 10 MHz < fx ≤ 20 MHz

EXCLK	OSCSEL	0	OSCSELS	0	0	0	AMPH
0	1	0	0/1	0	0	0	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 27 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	0	0	AMPH
1	1	0	0/1	0	0	0	×

Remarks 1. ×: don't care

2. For setting of the P123/XT1 and P124/XT2 pins, see **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 27 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	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN

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

Caution Be sure to clear bits 1 and 6 of PER0 register to 0.

Remark RTCEN: Control of the real-time counter input clock
ADCEN: Control of the A/D converter input clock
IIC0EN: Control of the serial interface IIC0 input clock
SAU1EN: Control of the serial array unit 1 input clock
SAU0EN: Control of the serial array unit 0 input clock
TAU0EN: Control of the timer array unit 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 17 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after STOP mode is released
If the X1 clock oscillates before the STOP mode is entered, set the value of the OSTS register before executing the STOP instruction.

<3> Executing the STOP instruction
When the STOP instruction is executed, the system is placed in the STOP mode and X1 oscillation is stopped (the input of the external clock is disabled).

(b) To stop X1 oscillation (disabling external clock input) by setting MSTOP to 1

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the high-speed system clock.

When CLS = 0 and MCS = 1, the high-speed system clock is supplied to the CPU, so change the CPU clock to the subsystem clock or internal high-speed oscillation clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Setting of X1 clock oscillation stabilization time after restart of X1 clock oscillation[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 17 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after STOP mode is released

If the X1 clock oscillates before the STOP mode is entered, set the value of the OSTS register before executing the STOP instruction.

<3> Executing the STOP instruction

When the STOP instruction is executed, the system is placed in the STOP mode and internal high-speed oscillation clock is stopped.

(b) To stop internal high-speed oscillation clock by setting HIOSTOP to 1

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the internal high-speed oscillation clock.

When CLS = 0 and MCS = 0, the internal high-speed oscillation clock is supplied to the CPU, so change the CPU clock to the high-speed system clock or subsystem clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the internal high-speed oscillation clock (CSC register)
When HIOSTOP is set to 1, internal high-speed oscillation clock is stopped.

Caution Be sure to confirm that MCS = 1 or CLS = 1 when setting HIOSTOP to 1. In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock.

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 IIC0 are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 27 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	0	0	AMPH
0/1	0/1	0	1	0	0	0	×

Remarks 1. ×: don't care

2. 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 IIC0 are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 27 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
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 (240 kHz (TYP.)) if the watchdog timer operation is enabled by the option byte.

The internal low-speed oscillator continues oscillation except when the watchdog timer stops. When the watchdog timer operates, the internal low-speed oscillation clock does not stop even in case of a program loop.

(1) Example of setting procedure when stopping the internal low-speed oscillation clock

The internal low-speed oscillation clock can be stopped in the following two ways.

- Stop the watchdog timer in the HALT/STOP mode by the option byte (bit 0 (WDSTBYON) of 000C0H = 0), and execute the HALT or STOP instruction.
- Stop the watchdog timer by the option byte (bit 4 (WDTON) of 000C0H = 0).

(2) Example of setting procedure when restarting oscillation of the internal low-speed oscillation clock

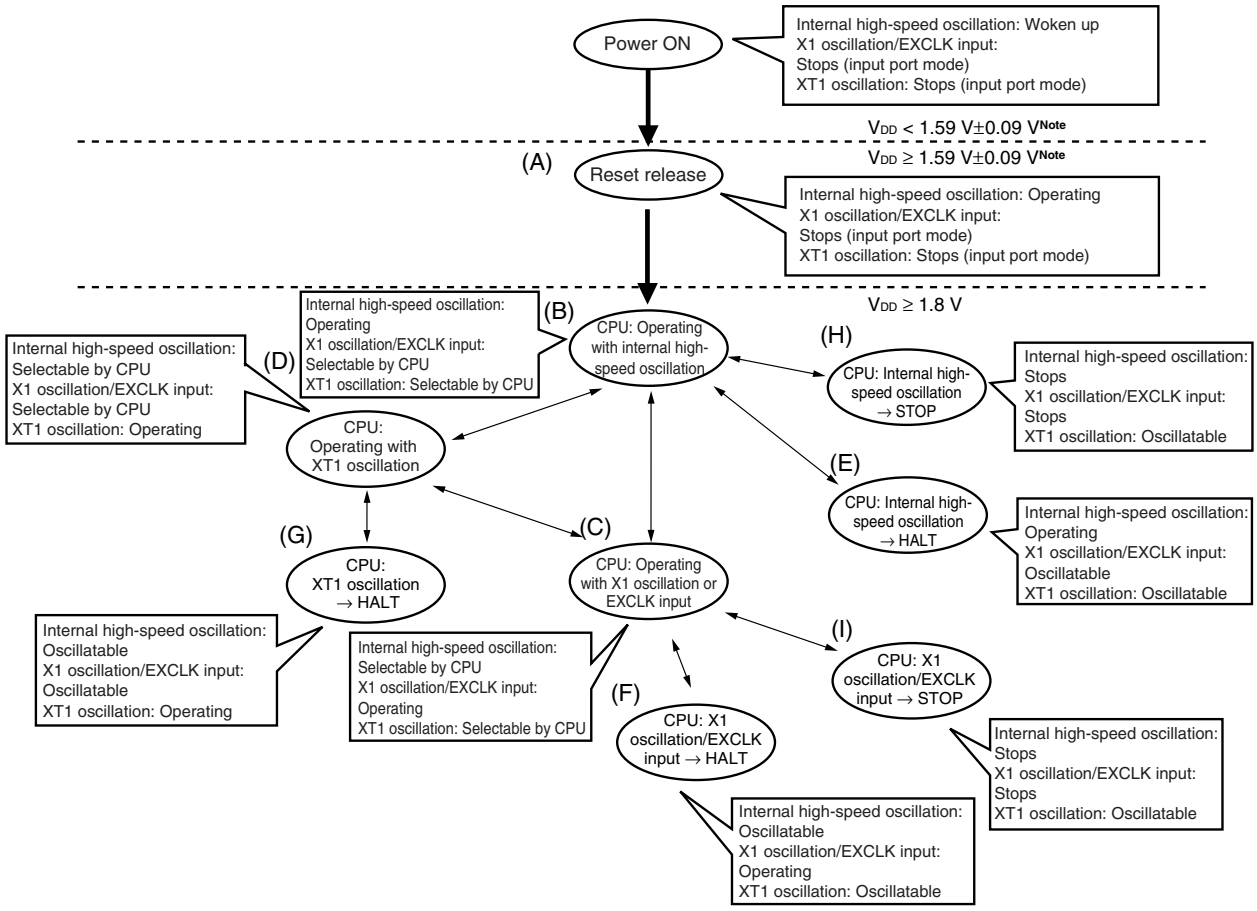
The internal low-speed oscillation clock can be restarted as follows.

- Release the HALT or STOP mode
  (only when the watchdog timer is stopped in the HALT/STOP mode by the option byte (bit 0 (WDSTBYON) of 000C0H) = 0) and when the watchdog timer is stopped as a result of execution of the HALT or STOP instruction).

5.6.5 CPU clock status transition diagram

Figure 5-15 shows the CPU clock status transition diagram of this product.

Figure 5-15. CPU Clock Status Transition Diagram



Note Preliminary value and subject to change.

Remark If the low-power-supply detector (LVI) is set to ON by default by the option bytes, the reset will not be released until the power supply voltage (V_{DD}) exceeds 2.07 V±0.2 V Note.

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

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 (1/4)

(1) CPU operating with internal high-speed oscillation clock (B) after reset release (A)

Status Transition	SFR Register Setting
(A) → (B)	SFR registers do not have to be set (default status after reset release).

(2) CPU operating with high-speed system clock (C) after reset release (A)

(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) ⟶

Setting Flag of SFR Register / Status Transition	CMC Register 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	0/1	0	0/1	Must not be checked	1

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

2. FSEL = 1 when f_{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 27 ELECTRICAL SPECIFICATIONS).

(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	XTSTOP		CSS
(A) → (B) → (D)	1	0	Necessary	1

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

Remark (A) to (I) in Table 5-4 correspond to (A) to (I) in Figure 5-15.

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

(4) CPU clock changing from internal high-speed oscillation clock (B) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[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	0/1	**Note 2**	0	0/1	Must not be checked	1

Unnecessary if these registers are already set

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

Notes 1. The CMC and OSMC registers 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 27 ELECTRICAL SPECIFICATIONS).

(5) CPU clock changing from internal high-speed oscillation clock (B) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[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.

Remark (A) to (I) in Table 5-4 correspond to (A) to (I) in Figure 5-15.

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

(6) CPU clock changing from high-speed system clock (C) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		MCM0
(C) → (B)	0	10 μs	0

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

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

(Setting sequence of SFR registers) →

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

Unnecessary if the CPU is operating with the subsystem clock

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

(8) CPU clock changing from subsystem clock (D) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	CKC Register	
	HIOSTOP	MCM0	CSS
(D) → (B)	0	0	0

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

Unnecessary if this register is already set

Remark (A) to (I) in Table 5-4 correspond to (A) to (I) in Figure 5-15.

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

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

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note 1]			OSTS Register	CSC Register	OSMC Register	OSTC Register	CKC Register	
	EXCLK	OSCSEL	AMPH		MSTOP	FSEL		MCM0	CSS
(D) → (C) (X1 clock: 2 MHz ≤ fx ≤ 10 MHz)	0	1	0	Note 2	0	0	Must be checked	1	0
(D) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	Note 2	0	1 [Note 3]	Must be checked	1	0
(D) → (C) (external main clock)	1	1	0/1	Note 2	0	0/1	Must not be checked	1	0
	Unnecessary if this register is already set			Unnecessary if the CPU is operating with the high-speed system clock				Unnecessary if these registers are already set	

Notes 1. The CMC and OSMC registers 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 27 ELECTRICAL SPECIFICATIONS).

(10) • HALT mode (E) set while CPU is operating with internal high-speed oscillation clock (B)
• HALT mode (F) set while CPU is operating with high-speed system clock (C)
• HALT mode (G) set while CPU is operating with subsystem clock (D)

Status Transition	Setting
(B) → (E) (C) → (F) (D) → (G)	Executing HALT instruction

(11) • STOP mode (H) set while CPU is operating with internal high-speed oscillation clock (B)
• STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence) →

Status Transition		Setting		
(B) → (H)	In X1 stop	Stopping peripheral functions that cannot operate in STOP mode	–	Executing STOP instruction
	In X1 oscillation		Sets the OSTS register	
(C) → (I)				

Remark (A) to (I) in Table 5-4 correspond to (A) to (I) in Figure 5-15.

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

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Internal high-speed oscillation clock	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	Operating current can be reduced by stopping internal high-speed oscillator (HIOSTOP = 1).
	External main system clock	Enabling input of external clock from EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	Subsystem clock	Stabilization of X1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	
X1 clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator • RSTOP = 0	X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	Subsystem clock	Stabilization of XT1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1).
External main system clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator • RSTOP = 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).
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	

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	Type
f_{IH}	◀⟷▶	f_{MX}	type 2 (see Table 5-8)
f_{MAIN}	◀⟷▶	f_{SUB}	type 3 (see Table 5-9)
f_{MAIN}	◀⟷▶ (changing the division ratio)	f_{MAIN}	type 1 (see Table 5-7)
f_{SUB}	◀⟷▶ (changing the division ratio)	f_{SUB}	type 1 (see Table 5-7)

Table 5-7. Maximum Number of Clocks Required in Type 1

Set Value Before Switchover	Set Value After Switchover	
	Clock A	Clock B
Clock A		1 + f_A/f_B clock
Clock B	1 + f_B/f_A clock	

Table 5-8. Maximum Number of Clocks Required in Type 2

Set Value Before Switchover		Set Value After Switchover	
MCM0		MCM0	
		0 ($f_{MAIN} = f_{IH}$)	1 ($f_{MAIN} = f_{MX}$)
0 ($f_{MAIN} = f_{IH}$)	$f_{MX} > f_{IH}$		1 + f_{MX}/f_{IH} clock
	$f_{MX} < f_{IH}$		$2f_{IH}/f_{MX}$ clock
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} > f_{IH}$	$2f_{MX}/f_{IH}$ 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 in Type 3

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

Remarks 1. The number of clocks listed in Table 5-7 to Table 5-9 is the number of CPU clocks before switchover.

2. 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

CHAPTER 6 TIMER ARRAY UNIT

The timer array unit has eight 16-bit timers per unit. 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 • Input pulse interval measurement • Measurement of high-/low-level width of input signal	• PWM output • One-shot pulse output • Multiple PWM output

Channel 7 can be used to realize LIN-bus reception processing in combination with UART3 of serial array unit 1.

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 (INTTM0n) at fixed intervals.

(2) Square wave output

A toggle operation is performed each time INTTM0n is generated and a square wave with a duty factor of 50% is output from a timer output pin (TO0k).

(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 (TI0k) has reached a specific value.

(4) Divider function

A clock input from a timer input pin (TI0k) is divided and output from an output pin (TO0k).

(5) Input pulse interval measurement

Counting is started by the valid edge of a pulse signal input to a timer input pin (TI0k). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.

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

Counting is started by a single edge of the signal input to the timer input pin (TI0k), 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 n: Channel number (n = 0 to 7), k: I/O port number (k = 0 to 6)

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 Modulator) output

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

(2) One-shot pulse output

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

(3) Multiple PWM (Pulse Width Modulator) output

By extending the PWM function and using one master channel and two or more slave channels, up to seven types of PWM signals that have a specific period and a specified duty factor can be generated.

6.1.3 LIN-bus supporting function (channel 7 only)

(1) Detection of wakeup signal

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD3) of UART3 and the count value of the timer is captured at the rising edge. In this way, a low-level width can be measured. If the low-level width is greater than a specific value, it is recognized as a wakeup signal.

(2) Detection of sync break field

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD3) of UART3 after a wakeup signal is detected, and the count value of the timer is captured at the rising edge. In this way, a low-level width is measured. If the low-level width is greater than a specific value, it is recognized as a sync break field.

(3) Measurement of pulse width of sync field

After a sync break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD3) of UART3 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

6.2 Configuration of Timer Array Unit

The timer array unit includes the following hardware.

Table 6-1. Configuration of Timer Array Unit

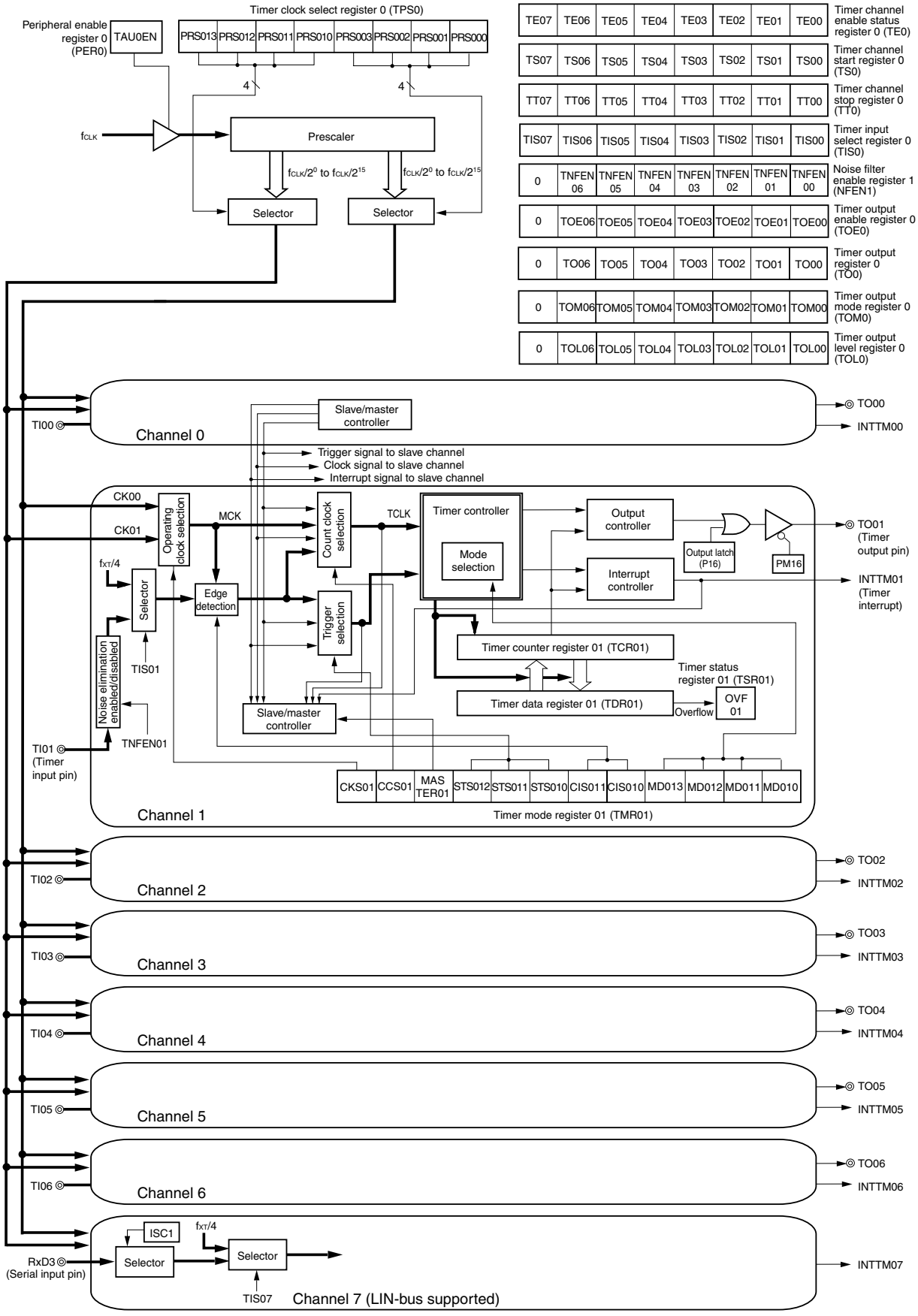
Item	Configuration
Timer/counter	Timer counter register 0n (TCR0n)
Register	Timer data register 0n (TDR0n)
Timer input	TI00 to TI06 pins, RxD3 pin (for LIN-bus)
Timer output	TO00 to TO06 pins, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register 0 (TPS0) • Timer channel enable status register 0 (TE0) • Timer channel start register 0 (TS0) • Timer channel stop register 0 (TT0) • Timer input select register 0 (TIS0) • Timer output enable register 0 (TOE0) • Timer output register 0 (TO0) • Timer output level register 0 (TOL0) • Timer output mode register 0 (TOM0)
	<Registers of each channel> • Timer mode register 0n (TMR0n) • Timer status register 0n (TSR0n) • Input switch control register (ISC) (channel 7 only) • Noise filter enable register 1 (NFEN1) • Port mode registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4) • Port registers 0, 1, 3, 4 (P0, P1, P3, P4)

Remark n: Channel number (n = 0 to 7)

Figure 6-1 shows the block diagram.

<R>

Figure 6-1. Block Diagram of Timer Array Unit



(1) Timer/counter register 0n (TCR0n)

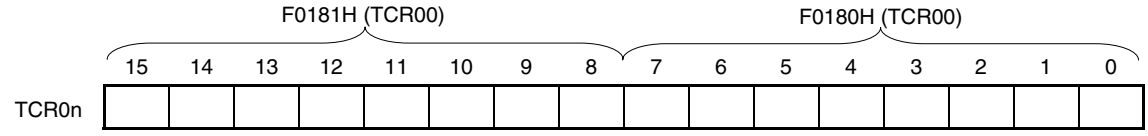
TCR0n is a 16-bit read-only register and is used to count clocks.

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 MD0n3 to MD0n0 bits of TMR0n.

Figure 6-2. Format of Timer/Counter Register 0n (TCR0n)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07) After reset: FFFFH R



(n = 0 to 7)

The count value can be read by reading TCR0n.

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

- When the reset signal is generated
- When the TAU0EN bit of peripheral enable register 0 (PER0) is cleared

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
- 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

Caution The count value is not captured to TDR0n even when TCR0n is read.

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

Table 6-2. TCR0n Register Read Value in Various Operation Modes

Operation Mode	Count Mode	TCR0n Register Read Value[Note]			
		Operation mode change after reset	Operation mode change after count operation paused (TT0n = 1)	Operation restart after count operation paused (TT0n = 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 TDR0n register + 1

Note The read values of the TCR0n register when TS0n has been set to "1" while TE0n = 0 are shown. The read value is held in the TCR0n register until the count operation starts.

Remark n = 0 to 7

(2) Timer data register 0n (TDR0n)

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 MD0n3 to MD0n0 bits of TMR0n.

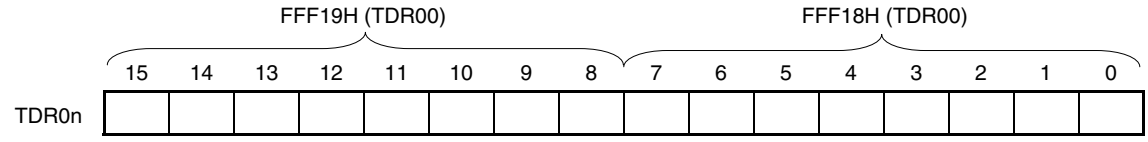
The value of TDR0n can be changed at any time.

This register can be read or written in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 6-3. Format of Timer Data Register 0n (TDR0n)

Address: FFF18H, FFF19H (TDR00), FFF1AH, FFF1BH (TDR01), FFF64H, FFF65H (TDR02) to FFF6EH, FFF6FH (TDR07) After reset: 0000H R/W



(n = 0 to 7)

(i) When TDR0n is used as compare register

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

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

(ii) When TDR0n is used as capture register

The count value of TCR0n is captured to TDR0n when the capture trigger is input.

A valid edge of the TI0k pin can be selected as the capture trigger. This selection is made by TMR0n.

Remark n = 0 to 7, k = 0 to 6

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register 0 (TPS0)
- Timer mode register 0n (TMR0n)
- Timer status register 0n (TSR0n)
- Timer channel enable status register 0 (TE0)
- Timer channel start register 0 (TS0)
- Timer channel stop register 0 (TT0)
- Timer input select register 0 (TIS0)
- Timer output enable register 0 (TOE0)
- Timer output register 0 (TO0)
- Timer output level register 0 (TOL0)
- Timer output mode register 0 (TOM0)
- Input switch control register (ISC)
- Noise filter enable register 1 (NFEN1)
- Port mode registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4)
- Port registers 0, 1, 3, 4 (P0, P1, P3, P4)

Remark n = 0 to 7

(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 is used, be sure to set bit 0 (TAU0EN) 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.

Cautions 1. When setting the timer array unit, be sure to set TAU0EN to 1 first. If TAU0EN = 0, writing to a control register of the timer array unit is ignored, and all read values are default values.

2. Be sure to clear bits 1 and 6 of PER0 register to 0.

Figure 6-4. 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	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN

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

(2) Timer clock select register 0 (TPS0)

TPS0 is a 16-bit register that is used to select two types of operation clocks (CK00, CK01) that are commonly supplied to each channel. CK01 is selected by bits 7 to 4 of TPS0, and CK00 is selected by bits 3 to 0.

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

Rewriting of PRS000 to PRS003 bits: Possible only when all the channels set to CKS0n = 0 are in the operation stopped state (TE0n = 0)

Rewriting of PRS010 to PRS013 bits: Possible only when all the channels set to CKS0n = 1 are in the operation stopped state (TE0n = 0)

TPS0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TPS0 can be set with an 8-bit memory manipulation instruction with TPS0L.

Reset signal generation clears this register to 0000H.

Figure 6-5. Format of Timer Clock Select Register 0 (TPS0)

Address: F01B6H, F01B7H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0	0	0	0	0	0	0	0	0	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000

PRS 0m3	PRS 0m2	PRS 0m1	PRS 0m0	Selection of operation clock (CK0m) 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 (TT0 = 00FFH).

Caution Be sure to clear bits 15 to 8 to "0".

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

2. m = 0, 1 n = 0 to 7

(3) Timer mode register 0n (TMR0n)

TMR0n sets an operation mode of channel n. It is used to select an operation clock (MCK), a count clock, whether the timer operates as the master or a slave, a start trigger and a capture trigger, the valid edge of the timer input, and an operation mode (interval, capture, event counter, one-count, or capture & one-count).

Rewriting TMR0n is prohibited when the register is in operation (when TE0 = 1). However, bits 7 and 6 (CIS0n1, CIS0n0) can be rewritten even while the register is operating with some functions (when TE0 = 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**).

TMR0n can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6-6. Format of Timer Mode Register 0n (TMR0n) (1/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS 0n	0	0	CCS 0n	MASTER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

CKS 0n	Selection of operation clock (MCK) of channel n
0	Operation clock CK00 set by PRS register
1	Operation clock CK01 set by PRS register
Operation clock MCK is used by the edge detector. A count clock (TCLK) is generated depending on the setting of the CCS0n bit.	

CCS 0n	Selection of count clock (TCLK) of channel n
0	Operation clock MCK specified by CKS0n bit
1	Valid edge of input signal input from TI0k pin
Count clock TCLK is used for the timer/counter, output controller, and interrupt controller.	

<R>

MASTER 0n	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
<R> 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 (MASTER0n = 1). Be sure to use the odd channel as a slave channel (MASTER0n = 0). Clear MASTER0n to 0 for a channel that is used with the single-operation function.	

Caution Be sure to clear bits 14, 13, 5, and 4 to "0".

Remark n = 0 to 7, k = 0 to 6

Figure 6-6. Format of Timer Mode Register 0n (TMR0n) (2/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS 0n	0	0	CCS 0n	MAST ER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

STS 0n2	STS 0n1	STS 0n0	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 TI0k pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of TI0k pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the combination-operation function).
Other than above			Setting prohibited

CIS 0n1	CIS 0n0	Selection of TI0k pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STS0n2 to STS0n0 bits is other than 010B, set the CIS0n1 to CIS0n0 bits to 10B.		

Remark n = 0 to 7, k = 0 to 6

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

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS 0n	0	0	CCS 0n	MAST ER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

MD 0n3	MD 0n2	MD 0n1	MD 0n0	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 MD0n0 bits varies depending on each operation mode (see table below).						

Operation mode (Value set by the MD0n3 to MD0n1 bits (see table above))	MD 0n0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0)	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 (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]. 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

Note If the start trigger (TS0n = 1) is issued during operation, the counter is cleared, an interrupt is generated, and recounting is started.

Remark n = 0 to 7

(4) Timer status register 0n (TSR0n)

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

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

TSR0n can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TSR0n can be set with an 8-bit memory manipulation instruction with TSR0nL.

Reset signal generation clears this register to 0000H.

Figure 6-7. Format of Timer Status Register 0n (TSR0n)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSR0n	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.	

Table 6-3. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVF	Set/clear conditions
• Capture mode • Capture & one-count mode	clear	When no overflow has occurred upon capturing
	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	–
	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 0 (TE0)

TE0 is used to enable or stop the timer operation of each channel.

When a bit of timer channel start register 0 (TS0) is set to 1, the corresponding bit of this register is set to 1.

When a bit of timer channel stop register 0 (TT0) is set to 1, the corresponding bit of this register is cleared to 0.

TE0 can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TE0 can be set with a 1-bit or 8-bit memory manipulation instruction with TE0L.

Reset signal generation clears this register to 0000H.

Figure 6-8. Format of Timer Channel Enable Status Register 0 (TE0)

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

TE0n	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.

Remark n = 0 to 7

(6) Timer channel start register 0 (TS0)

TS0 is a trigger register that is used to clear a timer counter (TCR0n) and start the counting operation of each channel.

When a bit (TS0n) of this register is set to 1, the corresponding bit (TE0n) of timer channel enable status register 0 (TE0) is set to 1. TS0n is a trigger bit and cleared immediately when TE0n = 1.

TS0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TS0 can be set with a 1-bit or 8-bit memory manipulation instruction with TS0L.

Reset signal generation clears this register to 0000H.

Figure 6-9. Format of Timer Channel Start Register 0 (TS0)

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

TS0n	Operation enable (start) trigger of channel n
0	No trigger operation
1	TE0n is set to 1 and the count operation becomes enabled. The TCR0n 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 to "0".

Remarks 1. When the TS0 register is read, 0 is always read.

2. n = 0 to 7

Table 6-4. Operations from Count Operation Enabled State to TCR0n Count Start (1/2)

Timer operation mode	Operation when TS0n = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TS0n=1) until count clock generation. The first count clock loads the value of TDR0n to TCR0n 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 TS0n bit loads the value of TDR0n to TCR0n. The subsequent count clock performs count down operation. The external trigger detection selected by STS0n2 to STS0n0 bits in the TMR0n 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 TCR0n 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 TCR0n Count Start (2/2)

Timer operation mode	Operation when TS0n = 1 is set
• One-count mode	When TS0n = 0, writing 1 to TS0n 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 TDR0n to TCR0n 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 TS0n = 0, writing 1 to TS0n 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 TCR0n 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

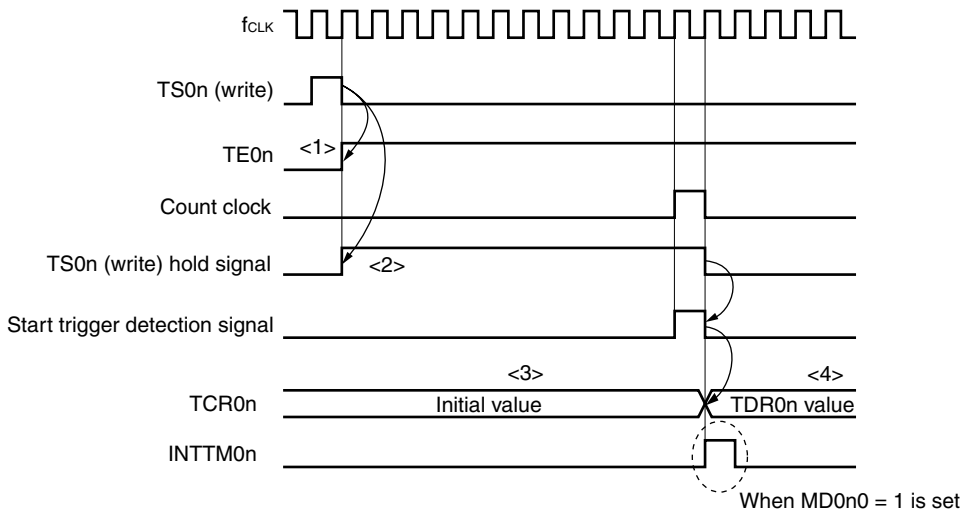
<1> Writing 1 to TS0n sets TE0n = 1

<2> The write data to TS0n is held until count clock generation.

<3> TCR0n holds the initial value until count clock generation.

<4> On generation of count clock, the “TDR0n value” is loaded to TCR0n and count starts.

Figure 6-10. Start Timing (In Interval Timer Mode)

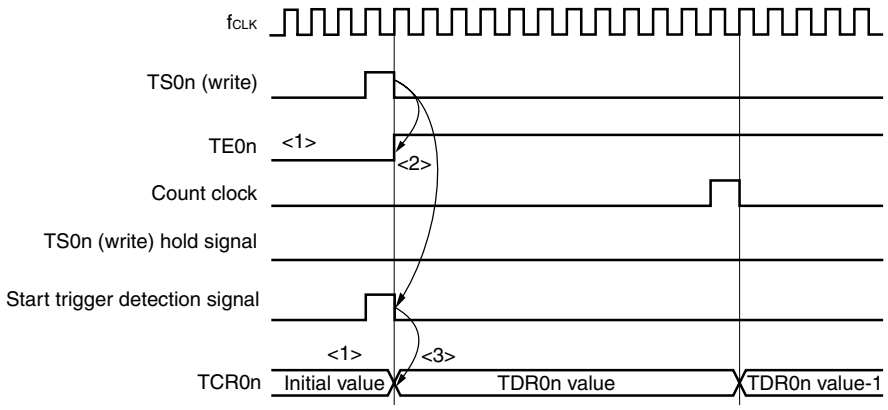


Caution In the first cycle operation of count clock after writing TS0n, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MD0n0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

(b) Start timing in event counter mode

<1> While TE0n is set to 0, TCR0n holds the initial value.
<2> Writing 1 to TS0n sets 1 to TE0n.
<3> As soon as 1 has been written to TS0n and 1 has been set to TE0n, the "TDR0n value" is loaded to TCR0n to start counting.
<4> After that, the TCR0n value is counted down according to the count clock.

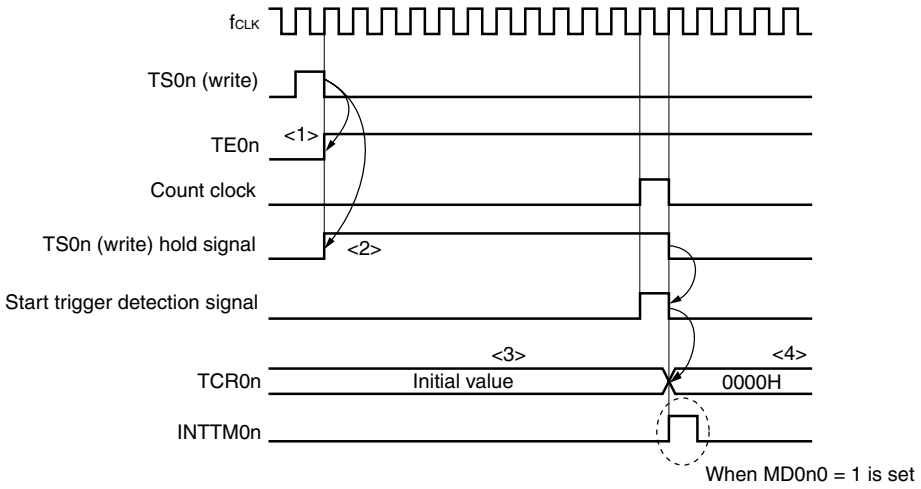
Figure 6-11. Start Timing (In Event Counter Mode)



(c) Start timing in capture mode

<1> Writing 1 to TS0n sets TE0n = 1
<2> The write data to TS0n is held until count clock generation.
<3> TCR0n holds the initial value until count clock generation.
<4> On generation of count clock, 0000H is loaded to TCR0n and count starts.

Figure 6-12. Start Timing (In Capture Mode)



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

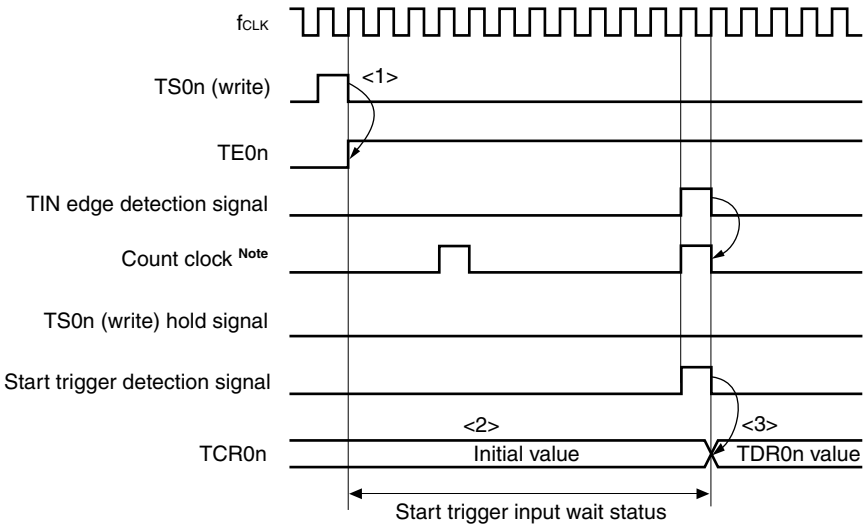
(d) Start timing in one-count mode

<1> Writing 1 to TS0n sets TE0n = 1

<2> Enters the start trigger input wait status, and TCR0n holds the initial value.

<3> On start trigger detection, the "TDR0n value" is loaded to TCR0n and count starts.

Figure 6-13. Start Timing (In One-count Mode)



Note When the one-count mode is set, the operation clock (MCK) is selected as count clock (CCS0n = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TI0k is used).

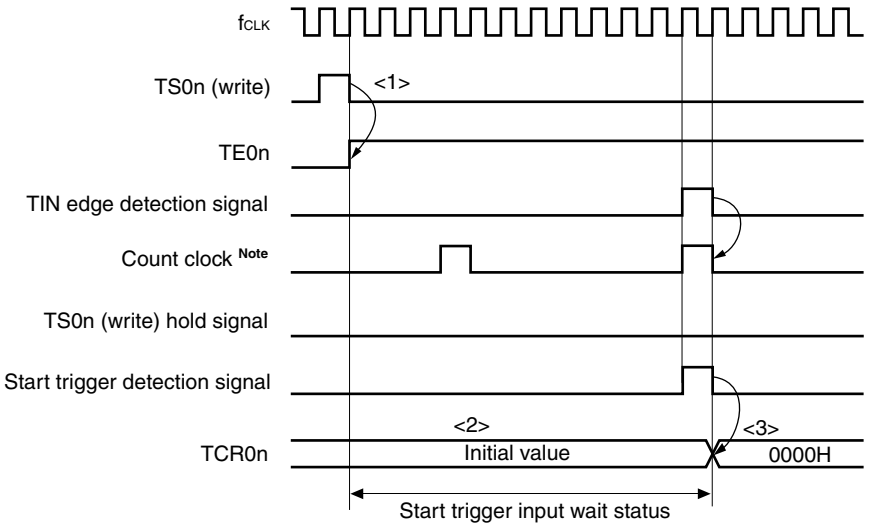
(e) Start timing in capture & one-count mode

<1> Writing 1 to TS0n sets TE0n = 1

<2> Enters the start trigger input wait status, and TCR0n holds the initial value.

<3> On start trigger detection, 0000H is loaded to TCR0n and count starts.

Figure 6-14. Start Timing (In Capture & One-count Mode)



Note When the capture & one-count mode is set, the operation clock (MCK) is selected as count clock (CCS0n = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TI0k is used).

(7) Timer channel stop register 0 (TT0)

TT0 is a trigger register that is used to clear a timer counter (TCR0n) and start the counting operation of each channel.

When a bit (TT0n) of this register is set to 1, the corresponding bit (TE0n) of timer channel enable status register 0 (TE0) is cleared to 0. TT0n is a trigger bit and cleared to 0 immediately when TE0n = 0.

TT0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TT0 can be set with a 1-bit or 8-bit memory manipulation instruction with TT0L.

Reset signal generation clears this register to 0000H.

Figure 6-15. Format of Timer Channel Stop Register 0 (TT0)

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

TT0n	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 to "0".

Remarks 1. When the TT0 register is read, 0 is always read.

2. n = 0 to 7

(8) Timer input select register 0 (TIS0)

TIS0 is used to select whether a signal input to the timer input pin (TI0n) or the subsystem clock divided by four (f_{XT}/4) is valid for each channel.

TIS0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

<R> **Figure 6-16. Format of Timer Input Select Register 0 (TIS0)**

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

TIS0n	Selection of timer input/subsystem clock used with channel n
0	Input signal of timer input pin (TI0n)
1	Subsystem clock divided by 4 (f_{XT}/4)

<R> **Caution Since the 78K0R/KE3 does not have the timer input pin on channel 7, normally the timer input on channel 7 cannot be used. When the LIN-bus communication function is used, select the input signal of the RxD3 pin by setting ISC1 (bit 1 of the input switch control register (ISC)) to 1 and setting TIS07 to 0.**

(9) Timer output enable register 0 (TOE0)

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

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

TOE0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOE0 can be set with a 1-bit or 8-bit memory manipulation instruction with TOE0L.

Reset signal generation clears this register to 0000H.

Figure 6-17. Format of Timer Output Enable Register 0 (TOE0)

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	0	TOE06	TOE05	TOE04	TOE03	TOE02	TOE01	TOE00

TOE0n	Timer output enable/disable of channel n
0	The TO0n operation stopped by count operation (timer channel output bit). Writing to the TO0n bit is enabled. The TO0n pin functions as data output, and it outputs the level set to the TO0n bit. The output level of the TO0n pin can be manipulated by software.
1	The TO0n operation enabled by count operation (timer channel output bit). Writing to the TO0n bit is disabled (writing is ignored). The TO0n pin functions as timer output, and the TOE0n is set or reset depending on the timer operation. The TO0n pin outputs the square-wave or PWM depending on the timer operation.

Caution Be sure to clear bits 15 to 7 to "0".

Remark n = 0 to 6

(10) Timer output register 0 (TO0)

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

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

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

<R> To use the P01/TO00, P16/TO01, P17/TO02, P31/TO03, P42/TO04, P05/TO05, or P06/TO06 pin as a port function pin, set the corresponding TO0n bit to “0”.

TO0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TO0 can be set with an 8-bit memory manipulation instruction with TO0L.

Reset signal generation clears this register to 0000H.

Figure 6-18. Format of Timer Output Register 0 (TO0)

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	0	TO06	TO05	TO04	TO03	TO02	TO01	TO00

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

Caution Be sure to clear bits 15 to 7 to “0”.

Remark n = 0 to 6

(11) Timer output level register 0 (TOL0)

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

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

TOL0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOL0 can be set with an 8-bit memory manipulation instruction with TOL0L.

Reset signal generation clears this register to 0000H.

Figure 6-19. Format of Timer Output Level Register 0 (TOL0)

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	0	TOL06	TOL05	TOL04	TOL03	TOL02	TOL01	TOL00

TOL0n	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 7 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. n = 0 to 6

(12) Timer output mode register 0 (TOM0)

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

<R> When a channel is used for the single-operation function, set the corresponding bit of the channel to be used to 0.

<R> 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 (TOE0n = 1).

TOM0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOM0 can be set with an 8-bit memory manipulation instruction with TOM0L.

Reset signal generation clears this register to 0000H.

<R> **Figure 6-20. Format of Timer Output Mode Register 0 (TOM0)**

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	0	TOM06	TOM05	TOM04	TOM03	TOM02	TOM01	TOM00

TOM0n	Control of timer output mode of channel n
0	Toggle mode (to produce toggle output by timer interrupt request signal (INTTM0n))
1	Combination-operation mode (output is set by the timer interrupt request signal (INTTM0n) of the master channel, and reset by the timer interrupt request signal (INTTM0m) of the slave channel)

Caution Be sure to clear bits 15 to 7 to "0".

Remark n: Channel number, m: Slave channel number
n = 0 to 6 (n = 0, 2, 4 for master channel)
n < m ≤ 6 (where m is a consecutive integer greater than n)

(13) Input switch control register (ISC)

ISC is used to implement LIN-bus communication operation with channel 7 in association with serial array unit 1.

When bit 1 of this register is set to 1, the input signal of the serial data input pin (RxD3) is selected as a timer input signal.

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

Reset signal generation clears this register to 00H.

Figure 6-21. Format of Input Switch Control Register (ISC)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit
0	Not uses the input signal (normal operation).
1	Input signal of RxD3 pin is used as timer input (wakeup signal detection).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD3 pin as an external interrupt (to measure the pulse widths of the sync break field and sync field).

Caution Be sure to clear bits 7 to 2 to "0".

<R> **Remark** Since the 78K0R/KE3 does not have the timer input pin on channel 7, normally the timer input on channel 7 cannot be used. When the LIN-bus communication function is used, select the input signal of the RxD3 pin by setting ISC1 to 1 and setting TIS07 (bit 7 of the timer input select register 0 (TIS0)) to 0.

(14) Noise filter enable register 1 (NFEN1)

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

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

When the noise filter is ON, match detection and synchronization of the 2 clocks is performed with the CPU/peripheral hardware clock (f_{CLK}). When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock (f_{CLK}).

NFEN1 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	0	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN06	Enable/disable using noise filter of TI06/TO06/P06 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN05	Enable/disable using noise filter of TI05/TO05/P05 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN04	Enable/disable using noise filter of TI04/TO04/P42 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03/TO03/INTP4/P31 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02/TO02/P17 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01/TO01/INTP5/P16 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00/P00 pin input signal
0	Noise filter OFF
1	Noise filter ON

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

(15) Port mode registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4)

These registers set input/output of ports 0, 1, 3, and 4 in 1-bit units.

When using the P01/TO00, P05/TO05/TI05, P06/TO06/TI06, P16/TO01/TI01/INTP5, P17/TO02/TI02, P31/TO03/TI03/INTP4, and P42/TO04/TI04 pins for timer output, set PM01, PM05, PM06, PM16, PM17, PM31,and PM42 and the output latches of P01, P05, P06, P16, P17, P31, and P42 to 0.

When using the P00/TI00, P05/TO05/TI05, P06/TO06/TI06, P16/TO01/TI01/INTP5, P17/TO02/TI02, P31/TO03/TI03/INTP4, and P42/TO04/TI04 pins for timer input, set PM00, PM05, PM06, PM16, PM17, PM31, and PM42 to 1. At this time, the output latches of P00, P05, P06, P16, P17, P31, and P42 may be 0 or 1.

PM0, PM1, PM3, and PM4 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 6-23. Format of Port Mode Registers 0, 1, 3, 4 (PM0, PM1, PM3, PM4)

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

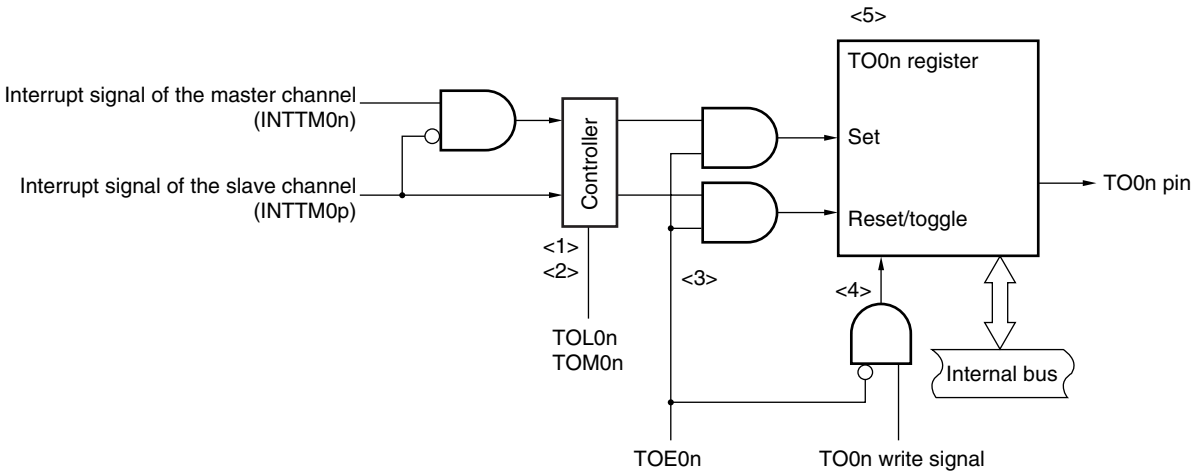
Symbol	7	6	5	4	3	2	1	0
PM4	1	1	1	1	PM43	PM42	PM41	PM40

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3, 4; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 Channel Output (TO0n pin) Control

6.4.1 TO0n pin output circuit configuration

Figure 6-24. Output Circuit Configuration



The following describes the TO0n pin output circuit.

<1> When TOM0n = 0 (toggle mode), the set value of the TOL0n register is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to the TO0n register.

<2> When TOM0n = 1 (combination-operation mode), both INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0n register.
At this time, the TOL0n register becomes valid and the signals are controlled as follows:

When TOL0n = 0: Forward operation (INTTM0 → set, INTTM0p → reset)
When TOL0n = 1: Reverse operation (INTTM0 → reset, INTTM0p → set)

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

<3> When TOE0n = 1, INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0n register. Writing to the TO0n register (TO0n write signal) becomes invalid.
When TOE0n = 1, the TO0n pin output never changes with signals other than interrupt signals.
To initialize the TO0n pin output level, it is necessary to set TOE0n = 0 and to write a value to TO0n.

<4> When TOE0n = 0, writing to TO0n bit to the target channel (TO0n write signal) becomes valid. When TOE0n = 0, neither INTTM0n (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to TO0n register.

<5> The TO0n register can always be read, and the TO0n pin output level can be checked.

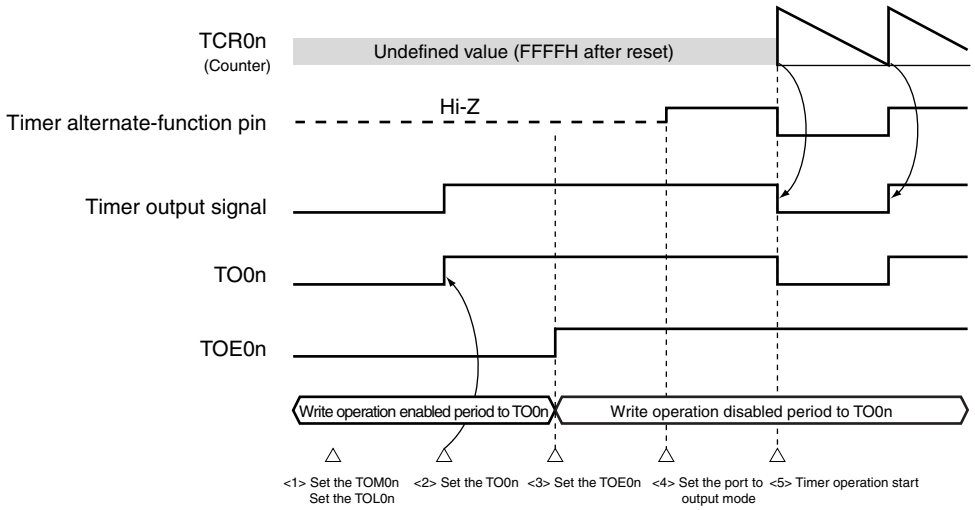
Remarks 1. n = 0 to 6 (n = 0, 2, or 4 for master channel)

2. p = n + 1, n + 2, n + 3 ... (where p ≤ 6)

6.4.2 TO0n Pin Output Setting

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

Figure 6-25. Status Transition from Timer Output Setting to Operation Start



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

- TOM0n bit (0: Toggle mode, 1: Combination-operation mode)
- TOL0n bit (0: Forward output, 1: Reverse output)

<2> The timer output signal is set to the initial status by setting TO0n.
<3> The timer output operation is enabled by writing 1 to TOE0n (writing to TO0n is disabled).
<4> The port I/O setting is set to output (see **6.3 (15) Port mode registers 0, 1, 3, 4**).
<5> The timer operation is enabled (TS0n = 1).

Remark n = 0 to 6

6.4.3 Cautions on Channel Output Operation

(1) Changing values set in registers TO0,TOE0,TOL0, and TOM0 during timer operation

Since the timer operations (operations of TCR0n and TDR0n) are independent of the TO0n output circuit and changing the values set in TO0, TOE0, TOL0, and TOM0 does not affect the timer operation, the values can be changed during timer operation.
When the values set in TOE0, TOL0, and TOM0 (except for TO0) are changed close to the timer interrupt (INTTM0n), the waveform output to the TO0n pin may be different depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTM0n) signal generation timing.

Remark n = 0 to 6

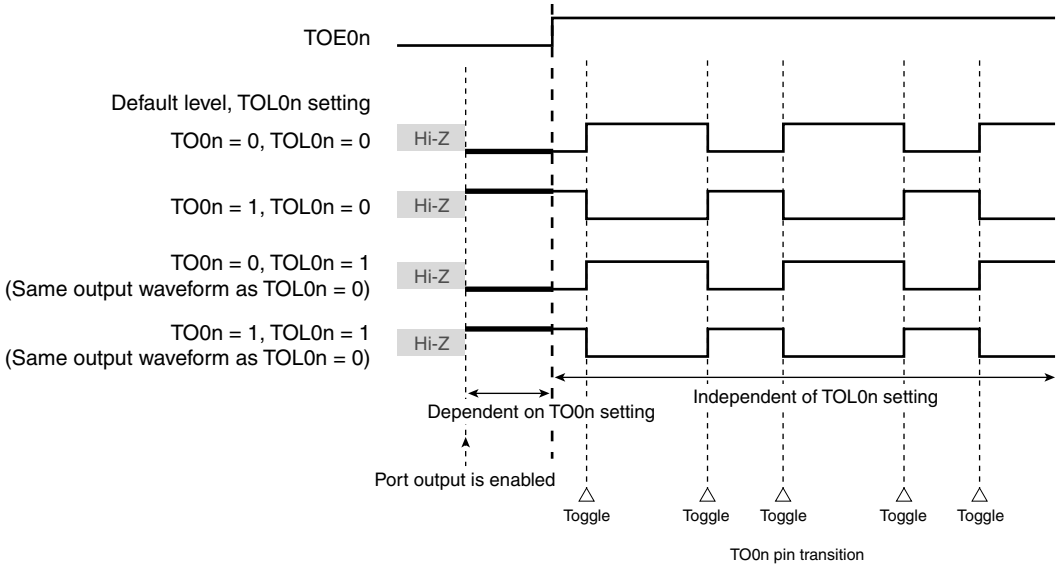
(2) Default level of TO0n pin and output level after timer operation start

The following figure shows the TO0n pin output level transition when writing has been done in the state of TOE0n = 0 before port output is enabled and TOE0n = 1 is set after changing the default level.

(a) When operation starts with TOM0n = 0 setting (toggle output)

The setting of TOL0n is invalid when TOM0n = 0. When the timer operation starts after setting the default level, the toggle signal is generated and the output level of TO0n pin is reversed.

Figure 6-26. TO0n Pin Output Status at Toggle Output (TOM0n = 0)



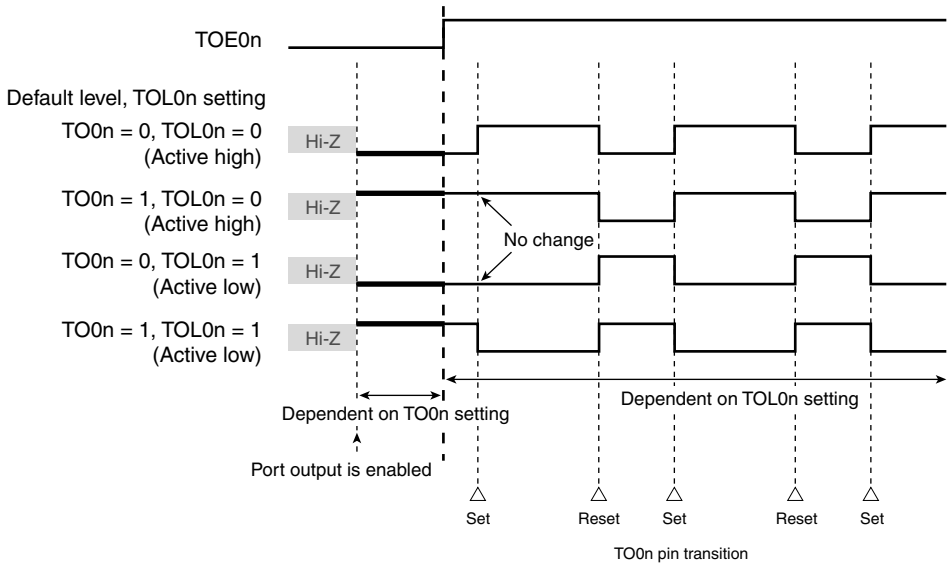
Remarks 1. Toggle: Reverse TO0n pin output status

2. n = 0 to 6

(b) When operation starts with TOM0n = 1 setting (Combination-operation mode (PWM output))

When TOM0n = 1, the active level is determined by TOL0n setting.

Figure 6-27. TO0n Pin Output Status at PWM Output (TOM0n = 1)



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

2. n = 0 to 6

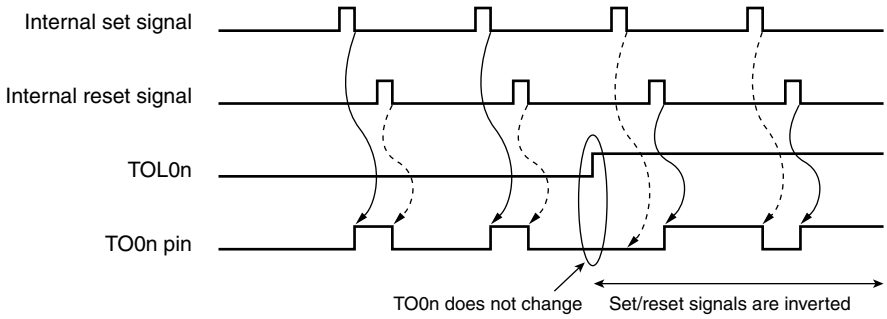
(3) Operation of TO0n pin in combination-operation mode (TOM0n = 1)

(a) When TOL0n setting has been changed during timer operation

When the TOL0n setting has been changed during timer operation, the setting becomes valid at the generation timing of TO0n change condition. Rewriting TOL0n does not change the output level of TO0n.

The following figure shows the operation when the value of TOL0n has been changed during timer operation (TOM0n = 1).

Figure 6-28. Operation when TOL0n Has Been Changed during Timer Operation



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

2. n = 0 to 6

(b) Set/reset timing

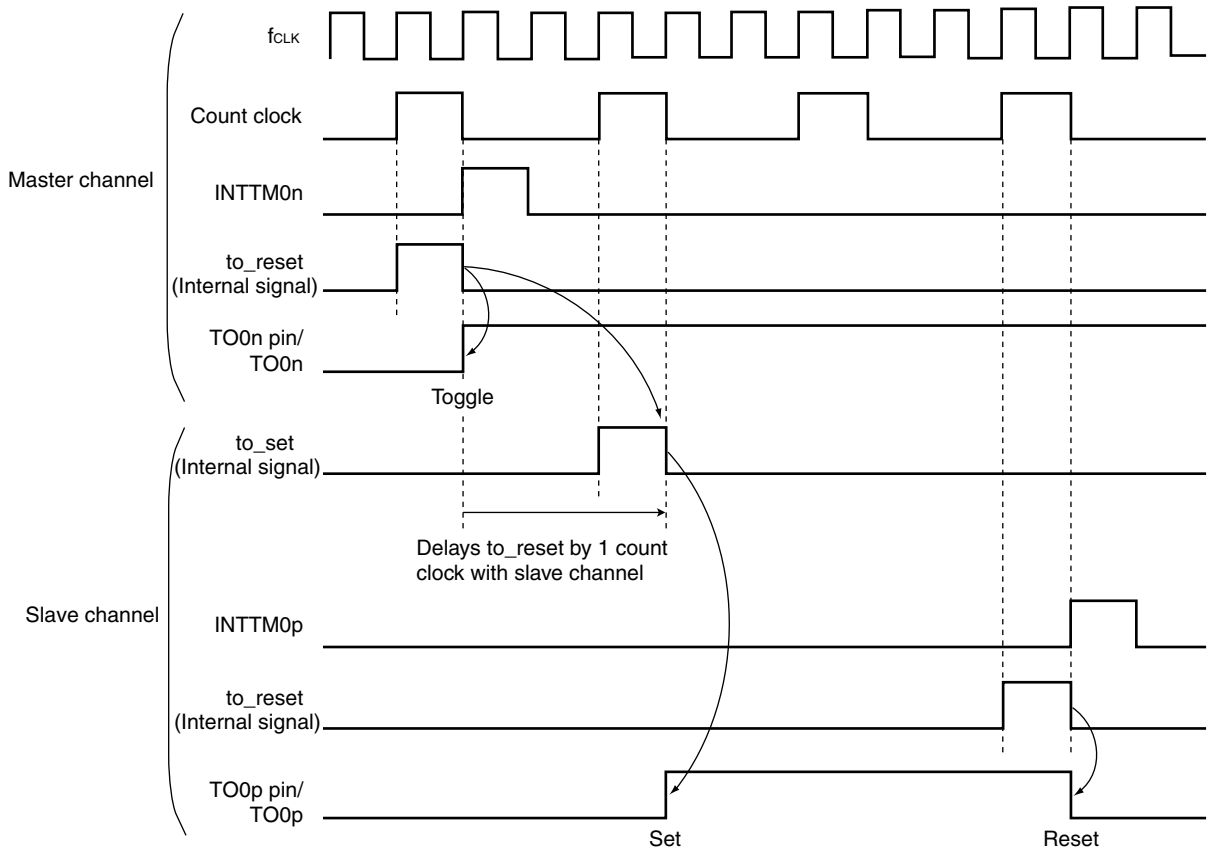
To realize 0%/100% output at PWM output, the TO0n pin/TO0n set timing at master channel timer interrupt (INTTM0n) generation is delayed by 1 count clock by the slave channel timer interrupt (INTTM0p).

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

Figure 6-29 shows the set/reset operating statuses where the master/slave channels are set as follows.

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

Figure 6-29. Set/Reset Timing Operating Statuses



Remarks 1. to_reset: TO0n pin reset/toggle signal
to_set: TO0n pin set signal

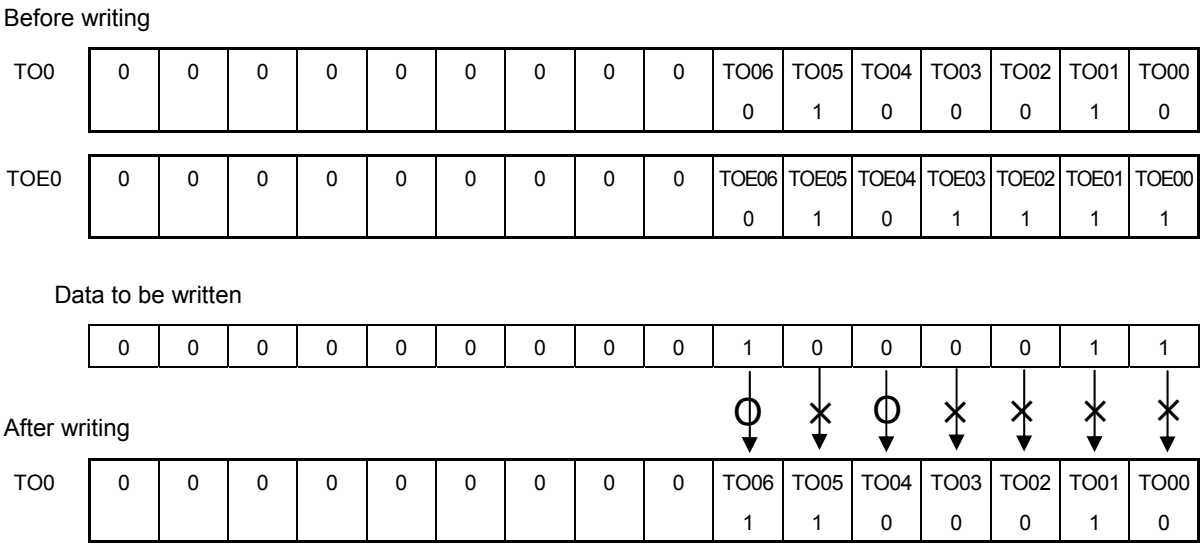
2. n = 0 to 6 (where n = 0, 2, or 4 for master channel)

3. p = n+1, n+2, n+3 ... (where p ≤ 6)

6.4.4 Collective manipulation of TO0n bits

In the TO0 register, the setting bits for all the channels are located in one register in the same way as the TS0 register (channel start trigger). Therefore, TO0n of all the channels can be manipulated collectively. Only specific bits can also be manipulated by setting the corresponding TOE0n = 0 to a target TO0n (channel output).

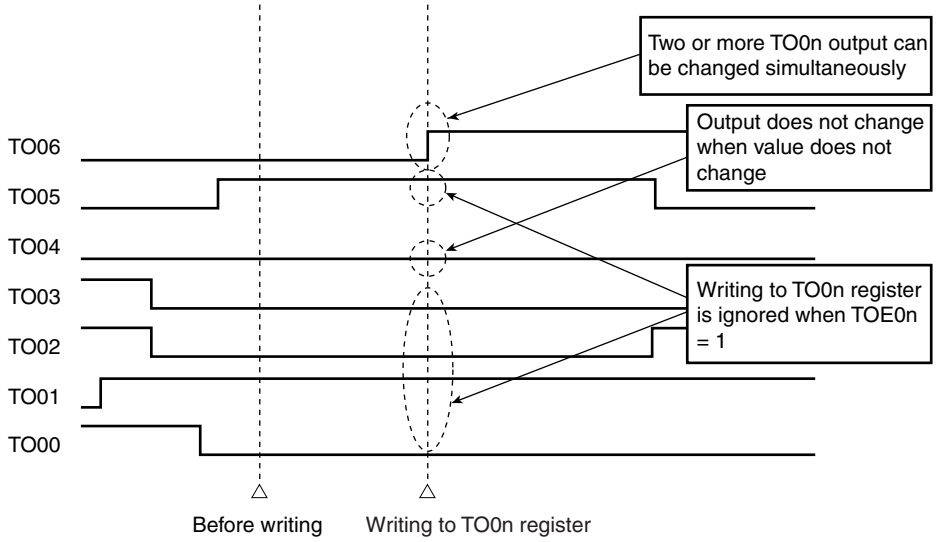
Figure 6-30. Example of TO0n Bits Collective Manipulation



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

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

Figure 6-31. TO0n Pin Statuses by Collective Manipulation of TOon Bits



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

Caution When TOE0n = 1, even if the output by timer interrupt of each timer (INTTM0n) contends with writing to TO0n, output is normally done to TO0n pin.

Remark n = 0 to 6

6.4.5 Timer Interrupt and TO0n Pin Output at Operation Start

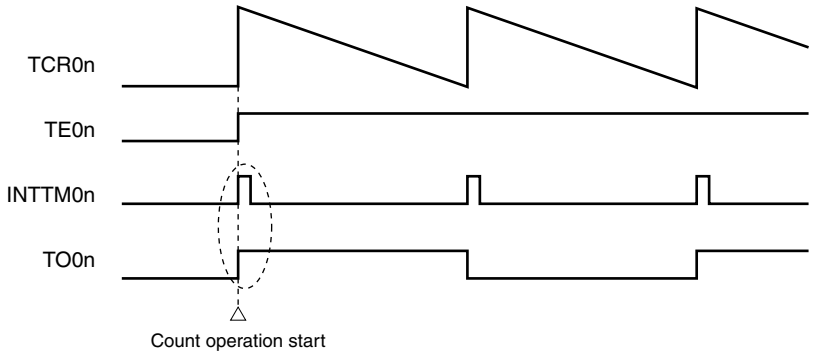
In the interval timer mode or capture mode, the MD0n0 bit in the TMR0n register sets whether or not to generate a timer interrupt at count start.

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

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

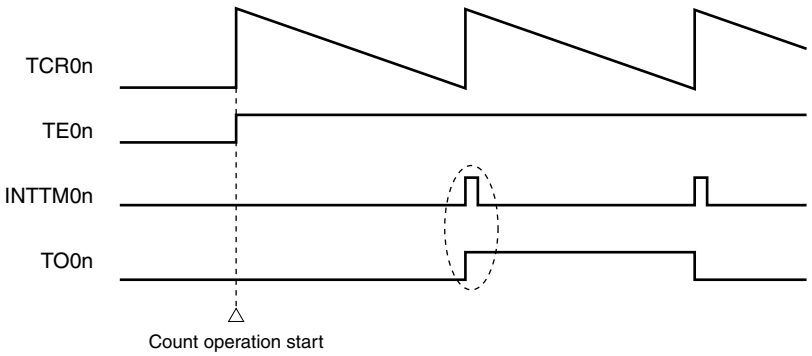
Figures 6-32 and 6-33 show operation examples when the interval timer mode (TOE0n = 1, TOM0n = 0) is set.

Figure 6-32. When MD0n0 is set to 1



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

Figure 6-33. When MD0n0 is set to 0



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

Remark n = 0 to 6

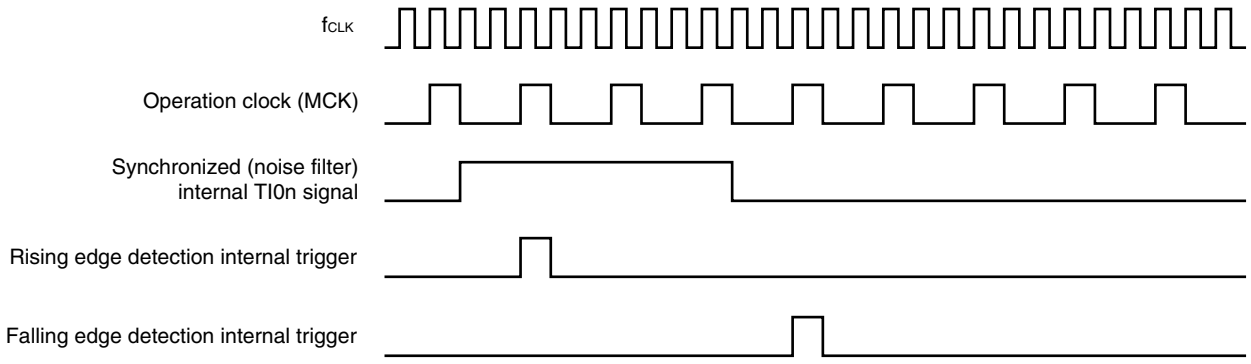
6.5 Channel Input (TI0n Pin) Control

6.5.1 TI0n edge detection circuit

(1) Edge detection basic operation timing

Edge detection circuit sampling is done in accordance with the operation clock (MCK).

Figure 6-34. Edge Detection Basic Operation Timing



Remark n = 0 to 6

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 is set as a master channel, channel 3 or those that follow (channels 3, 4, 5, etc.) can be set as a slave channel.

(4) Two or more slave channels can be set for one master channel.
(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example: If channels 0 and 4 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.

(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKS bit (bit 15 of the TMR0n 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 INTTM0n (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use the INTTM0n (interrupt), start software trigger, and count clock of the master channel, but it cannot transmit its own INTTM0n (interrupt), start software trigger, and count clock to the lower channel.
(9) A master channel cannot use the INTTM0n (interrupt), start software trigger, and count clock from the higher master channel.
(10) To simultaneously start channels that operate in combination, the TS0n bit of the channels in combination must be set at the same time.
(11) During a counting operation, the TS0n bit of all channels that operate in combination or only the master channel can be set. TS0n of only a slave channel cannot be set.
(12) To stop the channels in combination simultaneously, the TT0n bit of the channels in combination must be set at the same time.

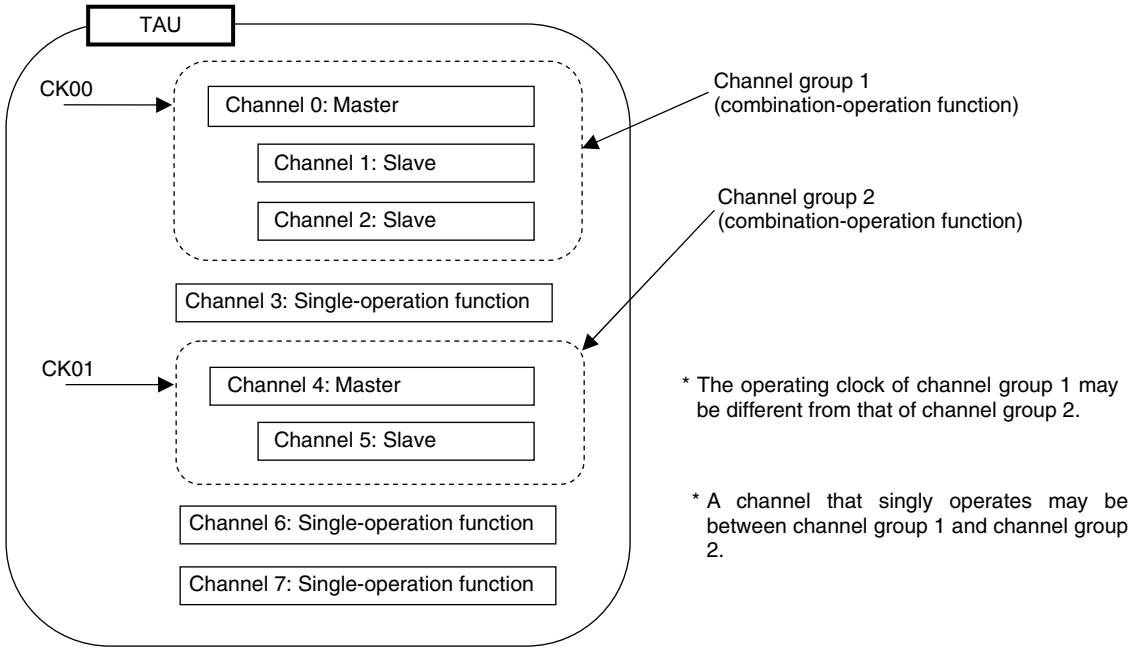
Remark n = 0 to 7

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 INTTM0n (timer interrupt) at fixed intervals.

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

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

(2) Operation as square wave output

TO0k performs a toggle operation as soon as INTTM0n has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TO0k can be calculated by the following expressions.

- Period of square wave output from TO0k = Period of count clock × (Set value of TDR0n + 1) × 2
- Frequency of square wave output from TO0k = Frequency of count clock/{(Set value of TDR0n + 1) × 2}

TCR0n operates as a down counter in the interval timer mode.

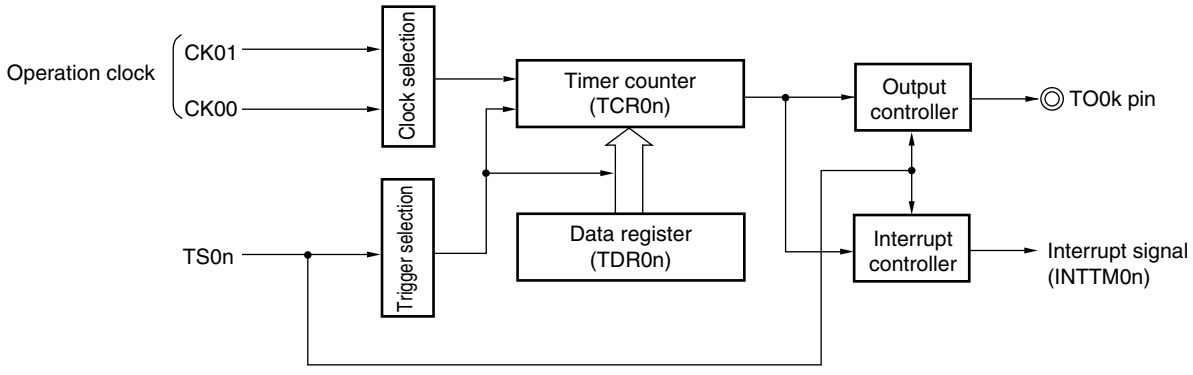
TCR0n loads the value of TDR0n at the first count clock after the channel start trigger bit (TS0n) is set to 1. If MD0n0 of TMR0n = 0 at this time, INTTM0n is not output and TO0k is not toggled. If MD0n0 of TMR0n = 1, INTTM0n is output and TO0k is toggled.

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

When TCR0n = 0000H, INTTM0n is output and TO0k is toggled at the next count clock. At the same time, TCR0n loads the value of TDR0n again. After that, the same operation is repeated.

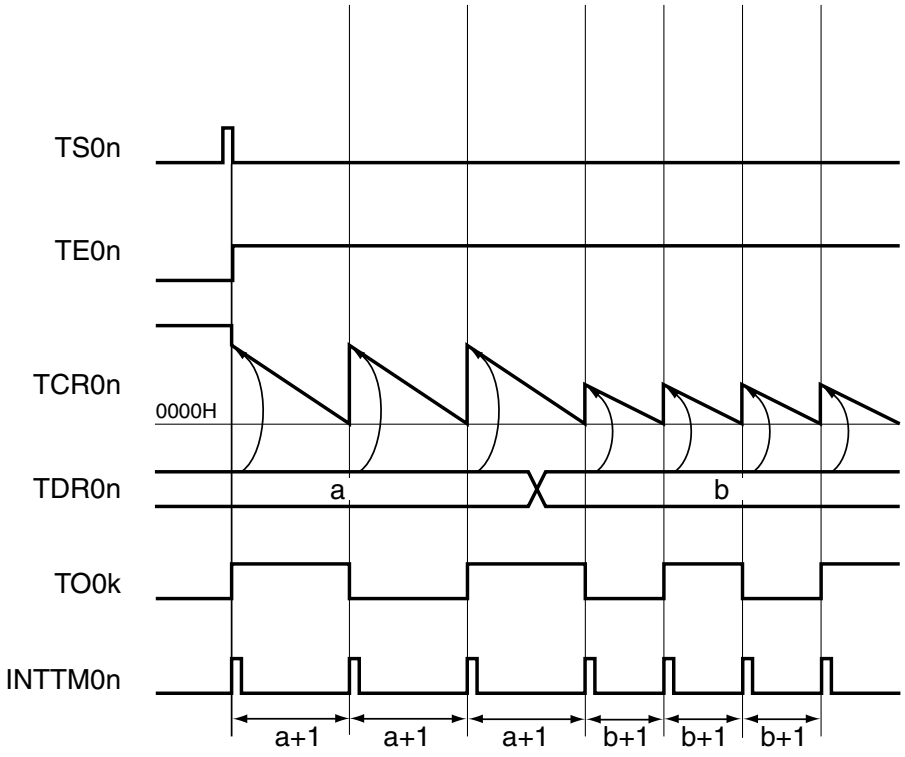
TDR0n can be rewritten at any time. The new value of TDR0n becomes valid from the next period.

Figure 6-35. Block Diagram of Operation as Interval Timer/Square Wave Output



Remark n = 0 to 7, k = 0 to 6

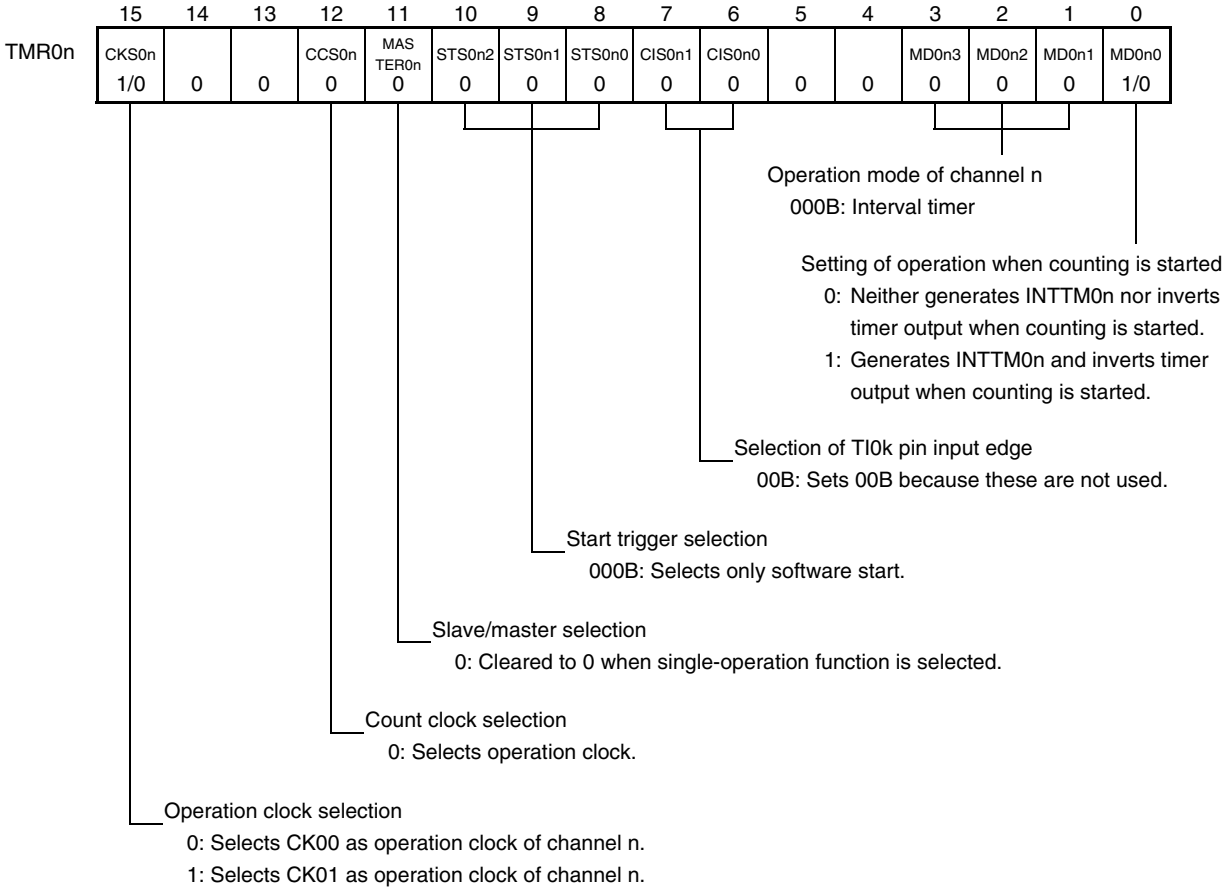
Figure 6-36. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MD0n0 = 1)



Remark n = 0 to 7, k = 0 to 6

Figure 6-37. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit k	
TO0	TO0k 1/0	0: Outputs 0 from TO0k. 1: Outputs 1 from TO0k.

(c) Timer output enable register 0 (TOE0)

	Bit k	
TOE0	TOE0k 1/0	0: Stops the TO0k output operation by counting operation. 1: Enables the TO0k output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit k	
TOL0	TOL0k 0	0: Cleared to 0 when TOM0k = 0 (toggle mode)

<R> **(e) Timer output mode register 0 (TOM0)**

	Bit k	
TOM0	TOM0k 0	0: Sets toggle mode.

Remark n = 0 to 7, k = 0 to 6

Figure 6-38. Operation Procedure of Interval Timer/Square Wave Output Function

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR0n register (determines operation mode of channel). Sets interval (period) value to the TDR0n register.	Channel stops operating. (Clock is supplied and some power is consumed.)
<R>	To use the TO0k output Clears the TOM0k bit of the TOM0 register to 0 (toggle mode). Clears the TOL0k bit to 0. Sets the TO0k bit and determines default level of the TO0k output. →	The TO0k pin goes into Hi-Z output state. The TO0k default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets TOE0k to 1 and enables operation of TO0k. →	TO0k does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0k pin outputs the TO0k set level.
Operation start	Sets TOE0k to 1 (only when operation is resumed). Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and count operation starts. Value of TDR0n is loaded to TCR0n at the count clock input. INTTM0n is generated and TO0k performs toggle operation if the MD0n0 bit of the TMR0n register is 1.
During operation	Set values of TMR0n, TOM0, and TOL0 registers cannot be changed. Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TO0 and TOE0 registers can be changed.	Counter (TCR0n) counts down. When count value reaches 0000H, the value of TDR0n is loaded to TCR0n again and the count operation is continued. By detecting TCR0n = 0000H, INTTM0n is generated and TO0k performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. TCR0n holds count value and stops. The TO0k output is not initialized but holds current status.
	TOE0k is cleared to 0 and value is set to TO0 register. →	The TO0k pin outputs the TO0k set level.
TAU stop	To hold the TO0k pin output level Clears TO0k bit to 0 after the value to be held is set to the port register. → When holding the TO0k pin output level is not necessary Switches the port mode register to input mode. →	The TO0k pin output level is held by port function. The TO0k pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0k bit is cleared to 0 and the TO0k pin is set to port mode.)

Operation is resumed.

Remark n = 0 to 7, k = 0 to 6

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 TI0k pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDR0n + 1

TCR0n operates as a down counter in the event counter mode.

When the channel start trigger bit (TS0n) is set to 1, TCR0n loads the value of TDR0n.

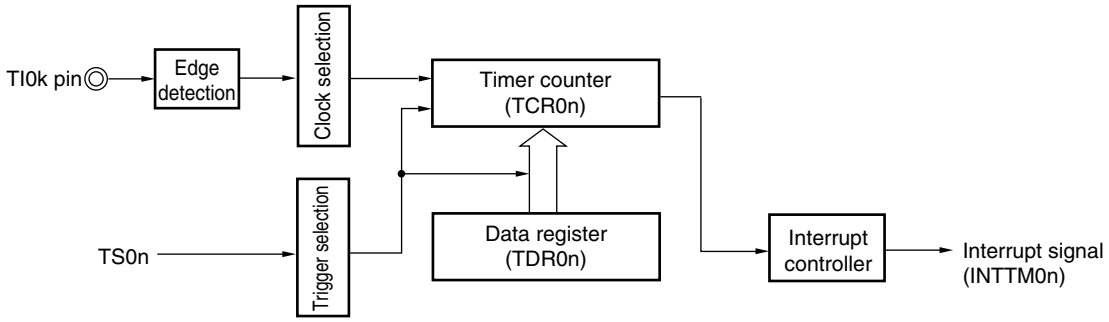
TCR0n counts down each time the valid input edge of the TI0k pin has been detected. When TCR0n = 0000H, TCR0n loads the value of TDR0n again, and outputs INTTM0n.

After that, the above operation is repeated.

TO0k must not be used because its waveform depends on the external event and irregular.

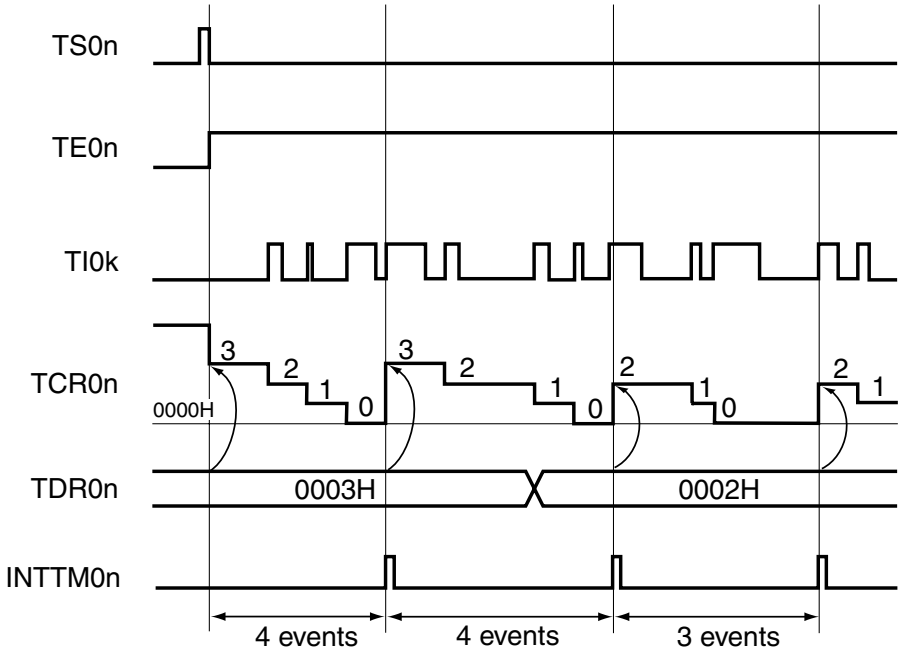
TDR0n can be rewritten at any time. The new value of TDR0n becomes valid during the next count period.

Figure 6-39. Block Diagram of Operation as External Event Counter



Remark n = 0 to 7, k = 0 to 6

Figure 6-40. Example of Basic Timing of Operation as External Event Counter



Remark n = 0 to 7, k = 0 to 6

Figure 6-41. Example of Set Contents of Registers in External Event Counter Mode

(a) Timer mode register 0n (TMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS0n			CCS0n	MASTER0n	STS0n2	STS0n1	STS0n0	CIS0n1	CIS0n0			MD0n3	MD0n2	MD0n1	MD0n0
	1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	1	1	0

Operation mode of channel n
011B: Event count mode

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

Selection of TI0k pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
1: Selects the TI0k pin input valid edge.

Operation clock selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit k
TO0	TO0k
	0

0: Outputs 0 from TO0k.

(c) Timer output enable register 0 (TOE0)

	Bit k
TOE0	TOE0k
	0

0: Stops the TO0k output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit k
TOL0	TOL0k
	0

0: Cleared to 0 when TOM0k = 0 (toggle mode).

<R> **(e) Timer output mode register 0 (TOM0)**

	Bit k
TOM0	TOM0k
	0

0: Sets toggle mode.

Remark n = 0 to 7, k = 0 to 6

Figure 6-42. Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of 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 TMR0n register (determines operation mode of channel). Sets number of counts to the TDR0n register. Clears the TOE0k bit of the TOE0 register to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and count operation starts. Value of TDR0n is loaded to TCR0n and detection of the TI0k pin input edge is awaited.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of TMR0n, TOM0, TOL0, TO0, and TOE0 registers cannot be changed.	Counter (TCR0n) counts down each time input edge of the TI0k pin has been detected. When count value reaches 0000H, the value of TDR0n is loaded to TCR0n again, and the count operation is continued. By detecting TCR0n = 0000H, the INTTM0n output is generated. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. TCR0n holds count value and stops.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n = 0 to 7, k = 0 to 6

6.7.3 Operation as frequency divider

The timer array unit can be used as a frequency divider that divides a clock input to the TI0k pin and outputs the result from TO0k.

The divided clock frequency output from TO0k can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDR0n + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDR0n + 1)

TCR0n operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS0n) is set to 1, TCR0n loads the value of TDR0n when the TI0k valid edge is detected. If MD0n0 of TMR0n = 0 at this time, INTTM0n is not output and TO0k is not toggled. If MD0n0 of TMR0n = 1, INTTM0n is output and TO0k is toggled.

After that, TCR0n counts down at the valid edge of TI0k. When TCR0n = 0000H, it toggles TO0k. At the same time, TCR0n loads the value of TDR0n again, and continues counting.

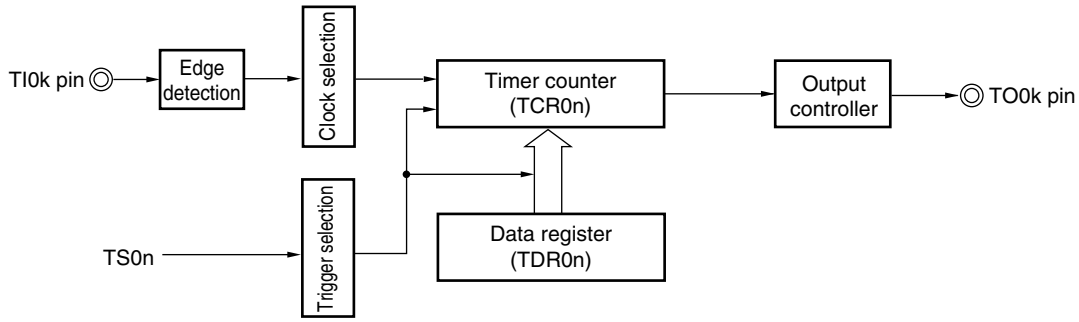
If detection of both the edges of TI0k is selected, the duty factor error of the input clock affects the divided clock period of the TO0k output.

The period of the TO0k output clock includes a sampling error of one period of the operation clock.

Clock period of TO0k output = Ideal TO0k output clock period ± Operation clock period (error)

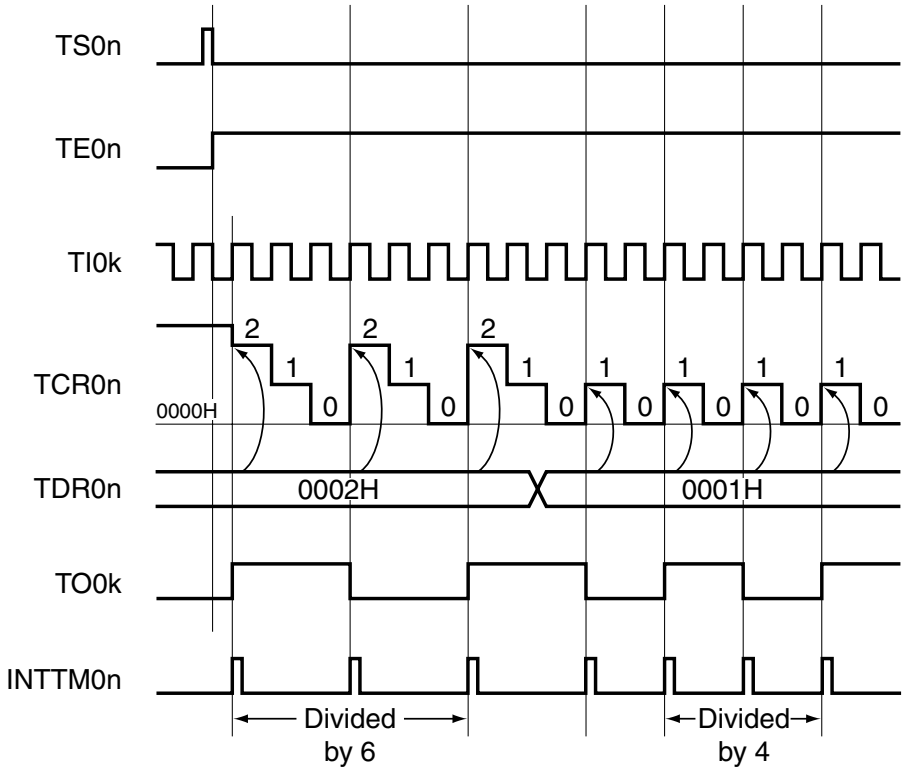
TDR0n can be rewritten at any time. The new value of TDR0n becomes valid during the next count period.

Figure 6-43. Block Diagram of Operation as Frequency Divider



Remark n = 0 to 7, k = 0 to 6

Figure 6-44. Example of Basic Timing of Operation as Frequency Divider (MD0n0 = 1)



Remark n = 0 to 7, k = 0 to 6

Figure 6-45. Example of Set Contents of Registers When Frequency Divider Is Used

(a) Timer mode register 0n (TMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS0n			CCS0n	MASTER0n	STS0n2	STS0n1	STS0n0	CIS0n1	CIS0n0			MD0n3	MD0n2	MD0n1	MD0n0
	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 INTTM0n nor inverts timer output when counting is started.
1: Generates INTTM0n and inverts timer output when counting is started.

Selection of TI0k pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when single-operation function is selected.

Count clock selection
1: Selects the TI0k pin input valid edge.

Operation clock selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit k
TO0	TO0k
	1/0

0: Outputs 0 from TO0k.
1: Outputs 1 from TO0k.

(c) Timer output enable register 0 (TOE0)

	Bit k
TOE0	TOE0k
	1/0

0: Stops the TO0k output operation by counting operation.
1: Enables the TO0k output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit k
TOL0	TOL0k
	0

0: Cleared to 0 when TOM0k = 0 (toggle mode)

<R> **(e) Timer output mode register 0 (TOM0)**

	Bit k
TOM0	TOM0k
	0

0: Sets toggle mode.

Remark n = 0 to 7, k = 0 to 6

Figure 6-46. Operation Procedure When Frequency Divider Function Is Used

<R>

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR0n register (determines operation mode of channel). Sets interval (period) value to the TDR0n register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOM0k bit of the TOM0 register to 0 (toggle mode). Clears the TOL0k bit to 0. Sets the TO0k bit and determines default level of the TO0k output. →	The TO0k pin goes into Hi-Z output state. The TO0k default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOE0k to 1 and enables operation of TO0k. →	TO0k does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0k pin outputs the TO0k set level.
Operation start	Sets the TOE0k to 1 (only when operation is resumed). Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and count operation starts. Value of TDR0n is loaded to TCR0n at the count clock input. INTTM0n is generated and TO0k performs toggle operation if the MD0n0 bit of the TMR0n register is 1.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of TO0 and TOE0 registers can be changed. Set values of TMR0n, TOM0, and TOL0 registers cannot be changed.	Counter (TCR0n) counts down. When count value reaches 0000H, the value of TDR0n is loaded to TCR0n again, and the count operation is continued. By detecting TCR0n = 0000H, INTTM0n is generated and TO0k performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. TCR0n holds count value and stops. The TO0k output is not initialized but holds current status.
	TOE0k is cleared to 0 and value is set to the TO0 register. →	The TO0k pin outputs the TO0k set level.
TAU stop	To hold the TO0k pin output level Clears TO0k bit to 0 after the value to be held is set to the port register. →	The TO0k pin output level is held by port function.
	When holding the TO0k pin output level is not necessary Switches the port mode register to input mode. →	The TO0k pin output level goes into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0k bit is cleared to 0 and the TO0k pin is set to port mode).

Operation is resumed.

Remark n = 0 to 7, k = 0 to 6

6.7.4 Operation as input pulse interval measurement

The count value can be captured at the TI0k valid edge and the interval of the pulse input to TI0k can be measured. The pulse interval can be calculated by the following expression.

TI0k input pulse interval = Period of count clock × ((10000H × TSR0n: OVF) + (Capture value of TDR0n + 1))

Caution The TI0k pin input is sampled using the operating clock selected with the CKS0n bit of the TMR0n register, so an error equal to the number of operating clocks occurs.

TCR0n operates as an up counter in the capture mode.

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

When the TI0k pin input valid edge is detected, the count value is transferred (captured) to TDR0n and, at the same time, the counter (TCR0n) is cleared to 0000H, and the INTTM0n is output. If the counter overflows at this time, the OVF bit of the TSR0n register is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

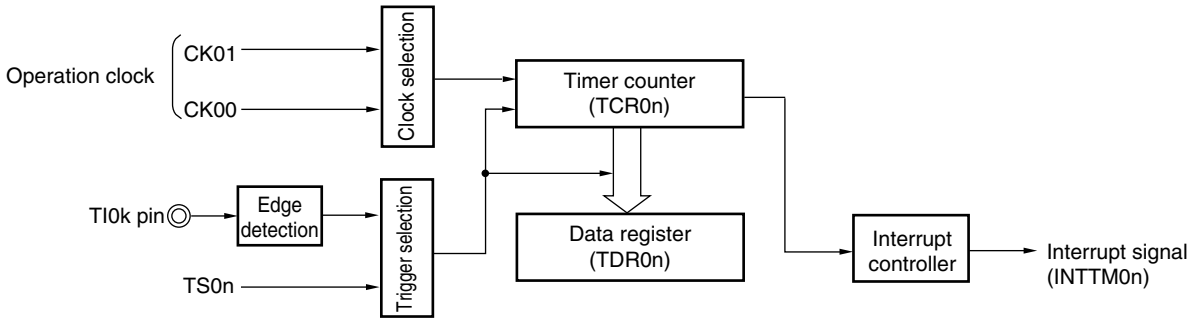
As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, the OVF bit is configured as a cumulative flag, the correct interval value cannot be measured if an overflow occurs more than once.

Set STS0n2 to STS0n0 of the TMR0n register to 001B to use the valid edges of TI0k as a start trigger and a capture trigger.

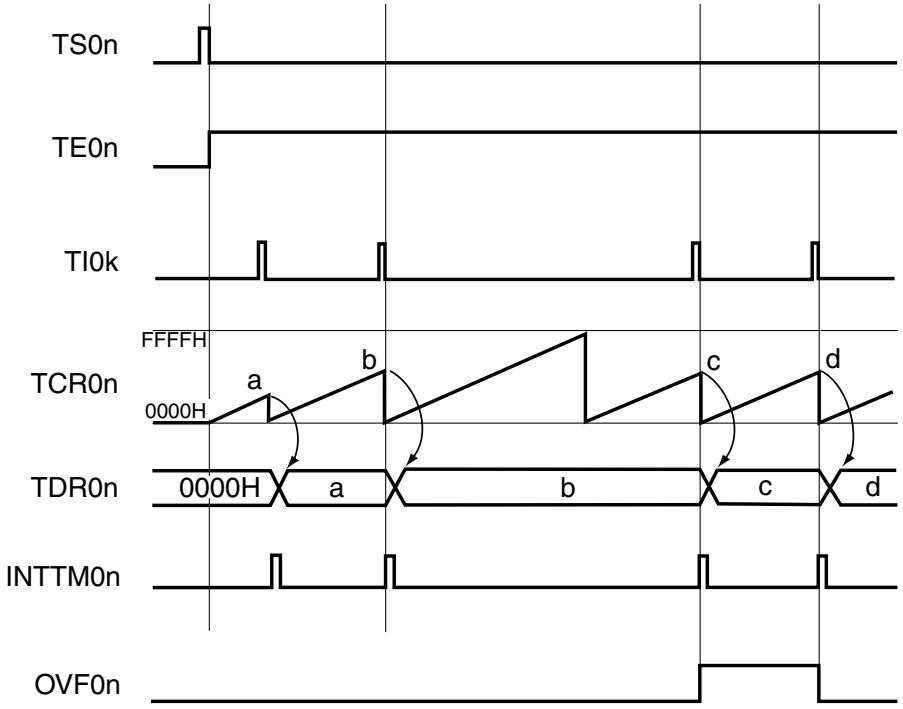
When TE0n = 1, instead of the TI0k pin input, a software operation (TS0n = 1) can be used as a capture trigger.

Figure 6-47. Block Diagram of Operation as Input Pulse Interval Measurement



Remark n = 0 to 7, k = 0 to 6

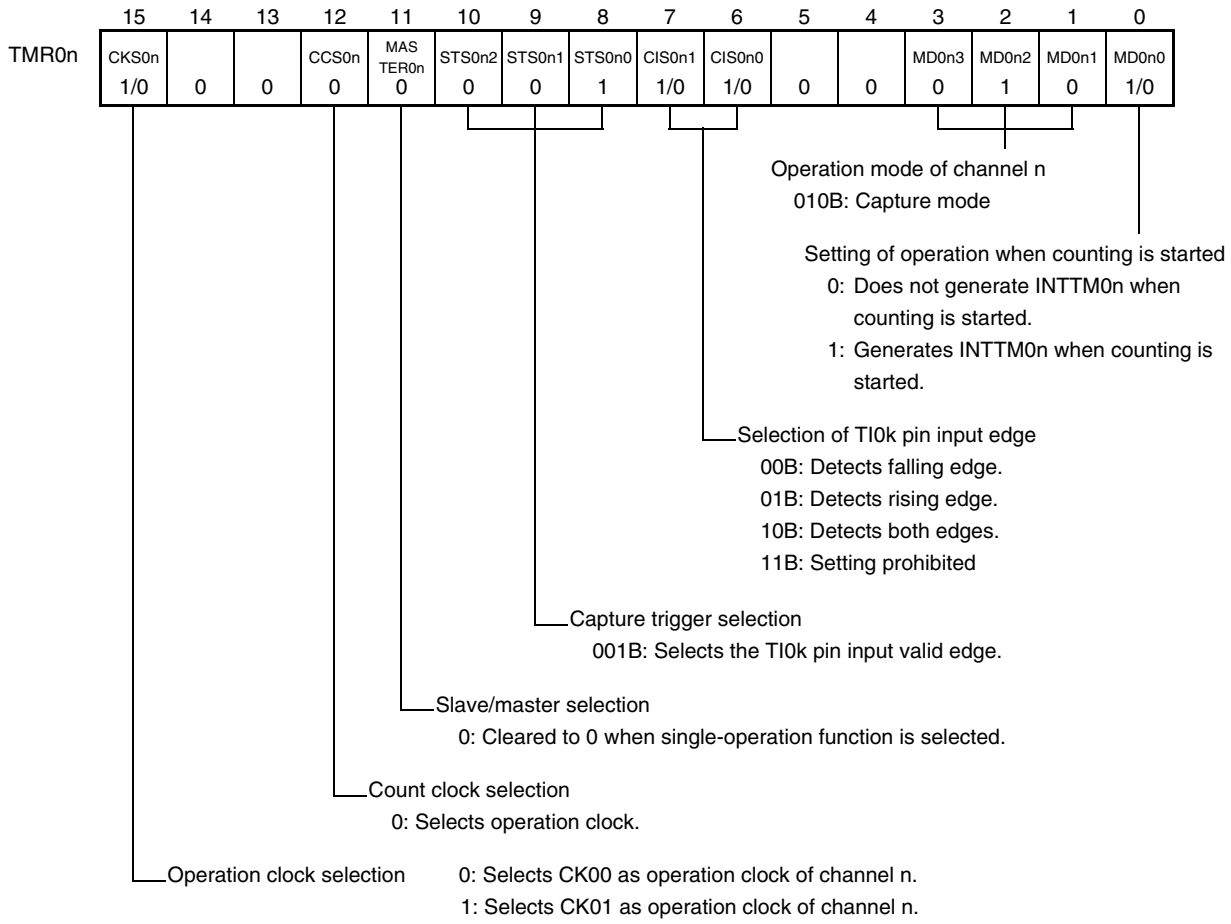
Figure 6-48. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MD0n0 = 0)



Remark n = 0 to 7, k = 0 to 6

Figure 6-49. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit k	
TO0	TO0k	0: Outputs 0 from TO0k.
	0	

(c) Timer output enable register 0 (TOE0)

	Bit k	
TOE0	TOE0k	0: Stops TO0k output operation by counting operation.
	0	

(d) Timer output level register 0 (TOL0)

	Bit k	
TOL0	TOL0k	0: Cleared to 0 when TOM0k = 0 (toggle mode).
	0	

<R> **(e) Timer output mode register 0 (TOM0)**

	Bit k	
TOM0	TOM0k	0: Sets toggle mode.
	0	

Remark n = 0 to 7, k = 0 to 6

Figure 6-50. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of 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 TMR0n register (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and count operation starts. TCR0n is cleared to 0000H at the count clock input. When the MD0n0 bit of the TMR0n register is 1, INTTM0n is generated.
During operation	Set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. The TDR0n register can always be read. The TCR0n register can always be read. The TSR0n register can always be read. Set values of TOM0, TOL0, TO0, and TOE0 registers cannot be changed.	Counter (TCRn) counts up from 0000H. When the TI0k pin input valid edge is detected, the count value is transferred (captured) to TDR0n. At the same time, TCR0n is cleared to 0000H, and the INTTM0n signal is generated. If an overflow occurs at this time, the OVF bit of the TSR0n register is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. TCR0n holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark n = 0 to 7, k = 0 to 6

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

By starting counting at one edge of TI0k and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TI0k can be measured. The signal width of TI0k can be calculated by the following expression.

Signal width of TI0k input = Period of count clock × ((10000H × TSRn: OVF) + (Capture value of TDR0n + 1))

Caution The TI0k pin input is sampled using the operating clock selected with the CKS0n bit of the TMR0n register, so an error equal to the number of operating clocks occurs.

TCR0n operates as an up counter in the capture & one-count mode.

When the channel start trigger (TS0n) is set to 1, TE0n is set to 1 and the TI0k pin start edge detection wait status is set.

When the TI0k start valid edge (rising edge of TI0k 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 TI0k when the high-level width is to be measured) is detected later, the count value is transferred to TDR0n and, at the same time, INTTM0n is output. If the counter overflows at this time, the OVF bit of the TSR0n register is set to 1. If the counter does not overflow, the OVF bit is cleared. TCR0n stops at the value “value transferred to TDR0n + 1”, and the TI0k pin start edge detection wait status is set. After that, the above operation is repeated.

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

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, the OVF bit is configured as an integral flag, and the correct interval value cannot be measured if an overflow occurs more than once.

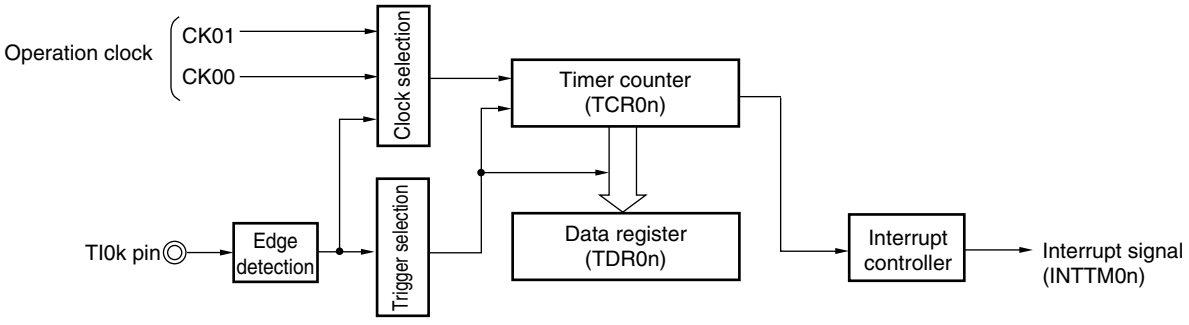
Whether the high-level width or low-level width of the TI0k pin is to be measured can be selected by using the CIS0n1 and CIS0n0 bits of the TMR0n register.

Because this function is used to measure the signal width of the TI0k pin input, TS0n cannot be set to 1 while TE0n is 1.

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

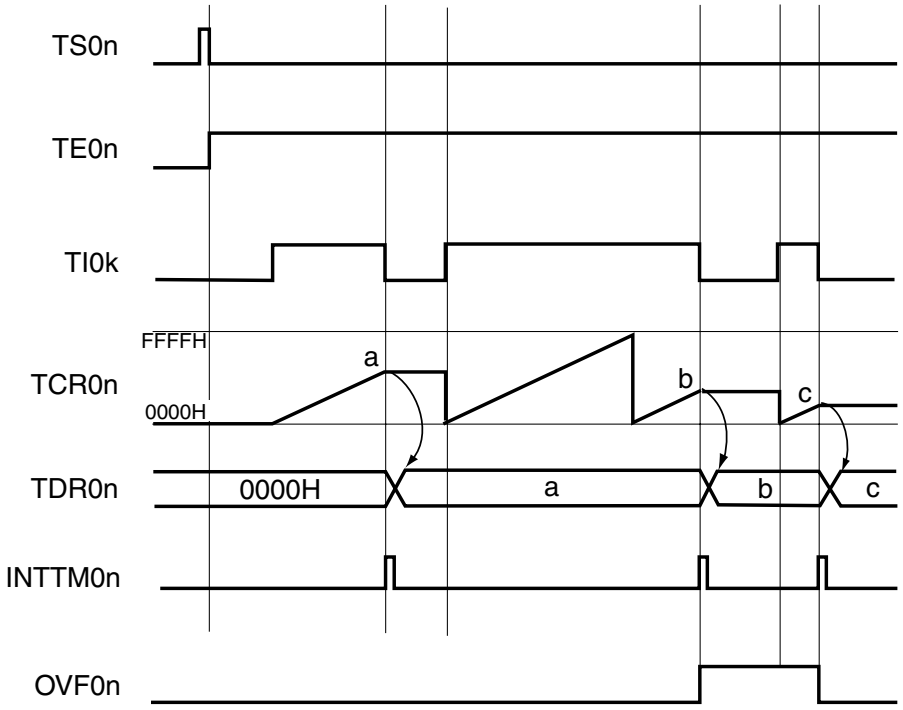
CIS0n1, CIS0n0 of TMR0n = 11B: High-level width is measured.

Figure 6-51. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Remark n = 0 to 7, k = 0 to 6

Figure 6-52. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remark n = 0 to 7, k = 0 to 6

Figure 6-53. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register 0n (TMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS0n			CCS0n	MASTER0n	STS0n2	STS0n1	STS0n0	CIS0n1	CIS0n0			MD0n3	MD0n2	MD0n1	MD0n0
	1/0	0	0	0	0	0	1	0	1	1/0	0	0	1	1	0	0

Operation mode of channel n
110B: Capture & one-count

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

Selection of TI0k pin input edge
10B: Both edges (to measure low-level width)
11B: Both edges (to measure high-level width)

Start trigger selection
010B: Selects the TI0k pin input valid edge.

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 CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit k
TO0	TO0k
	0

0: Outputs 0 from TO0k.

(c) Timer output enable register 0 (TOE0)

	Bit k
TOE0	TOE0k
	0

0: Stops the TO0k output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit k
TOL0	TOL0k
	0

0: Cleared to 0 when TOM0k = 0 (toggle mode).

<R> **(e) Timer output mode register 0 (TOM0)**

	Bit k
TOM0	TOM0k
	0

0: Sets toggle mode.

Remark n = 0 to 7, k = 0 to 6

Figure 6-54. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of 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 TMR0n register (determines operation mode of channel). Clears TOE0k to 0 and stops operation of TO0k.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0n bit to 1. → The TS0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 1, and the TI0k pin start edge detection wait status is set.
	Detects TI0k pin input count start valid edge. →	Clears TCR0n to 0000H and starts counting up.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of TMR0n, TOM0, TOL0, TO0, and TOE0 registers cannot be changed.	When the TI0k pin start edge is detected, the counter (TCRn) counts up from 0000H. If a capture edge of the TI0k pin is detected, the count value is transferred to TDR0n and INTTM0n is generated. If an overflow occurs at this time, the OVF bit of the TSR0n register is set; if an overflow does not occur, the OVF bit is cleared. TCR0n stops the count operation until the next TI0k pin start edge is detected.
Operation stop	The TT0n bit is set to 1. → TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. TCR0n holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	The TAU0EN 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 n = 0 to 7, k = 0 to 6

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 TDR0n (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDR0m (slave)}/{Set value of TDR0n (master) + 1} × 100
0% output: Set value of TDR0m (slave) = 0000H
100% output: Set value of TDR0m (slave) ≥ {Set value of TDR0n (master) + 1}

Remark The duty factor exceeds 100% if the set value of TDR0m (slave) > (set value of TDR0n (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode and counts the periods. When the channel start trigger (TS0n) is set to 1, INTTM0n is output. TCR0n counts down starting from the loaded value of TDR0n, in synchronization with the count clock. When TCR0n = 0000H, INTTM0n is output. TCR0n loads the value of TDR0n again. After that, it continues the similar operation.

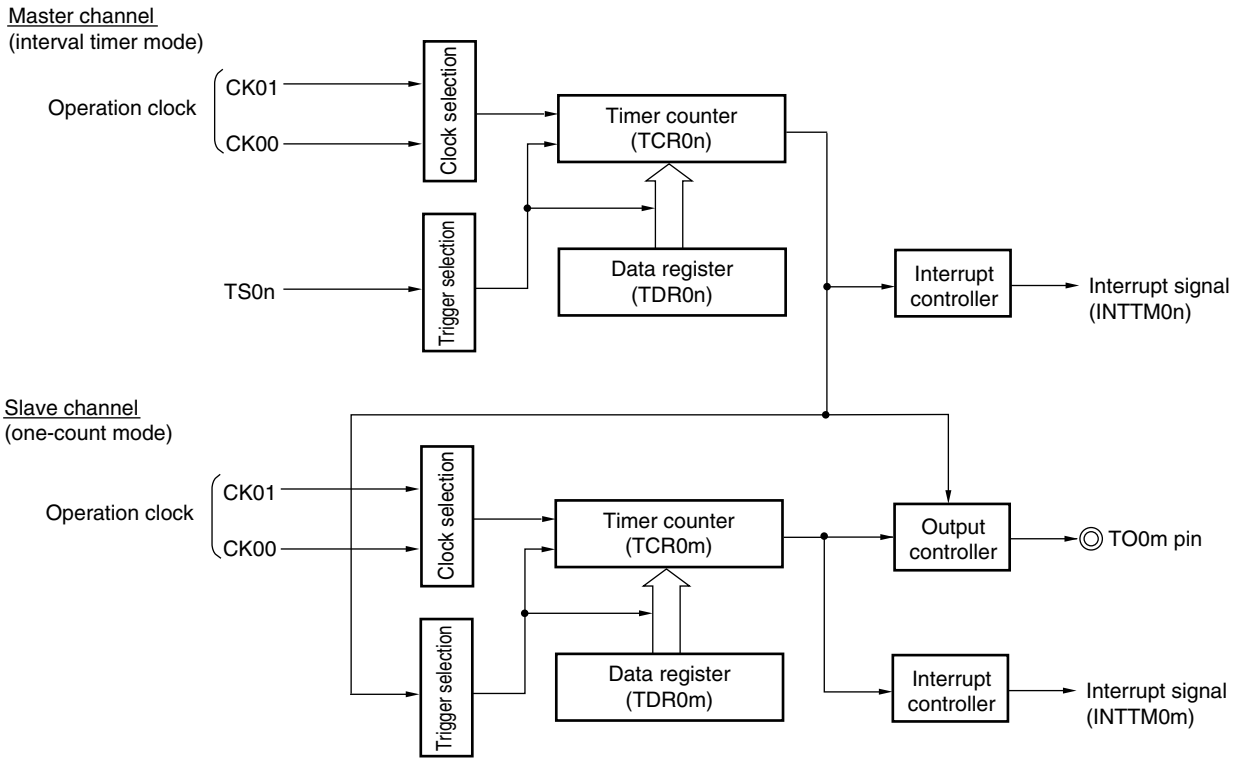
TCR0m of a slave channel operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0m pin. TCR0m of the slave channel loads the value of TDR0m, using INTTM0n of the master channel as a start trigger, and stops counting until the next start trigger (INTTM0n of the master channel) is input.

The output level of TO0m becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0m = 0000H.

Caution To rewrite both TDR0n of the master channel and TDR0m of the slave channel, a write access is necessary two times. The timing at which the values of TDR0n and TDR0m are loaded to TCR0n and TRC0m is upon occurrence of INTTM0n of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTM0n of the master channel, the TO0m pin cannot output the expected waveform. To rewrite both TDR0n of the master and TDR0m of the slave, therefore, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel.

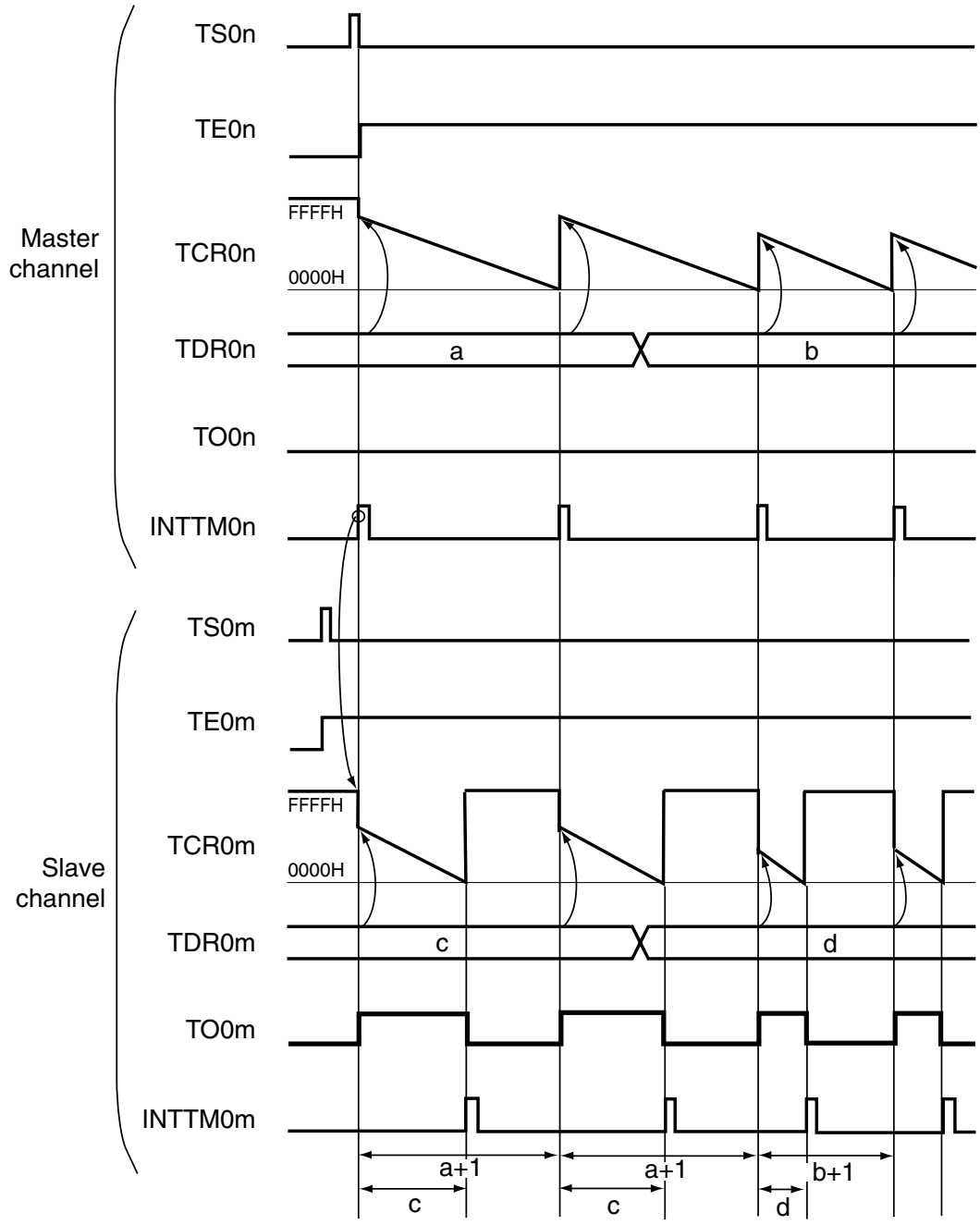
Remark n = 0, 2, 4
m = n + 1

Figure 6-55. Block Diagram of Operation as PWM Function



Remark n = 0, 2, 4
m = n + 1

Figure 6-56. Example of Basic Timing of Operation as PWM Function



Remark n = 0, 2, 4
m = n + 1

Figure 6-57. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register 0n (TMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS0n			CCS0n	MASTER0n	STS0n2	STS0n1	STS0n0	CIS0n1	CIS0n0			MD0n3	MD0n2	MD0n1	MD0n0
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTM0n when counting is started.

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

Start trigger selection
000B: Selects only software start.

Slave/master selection
1: Channel 1 is set as master channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CK00 as operation clock of channel n.
1: Selects CK01 as operation clock of channel n.

(b) Timer output register 0 (TO0)

	Bit n
TO0	TO0n
	0

0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOE0n
	0

0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOL0n
	0

0: Cleared to 0 when TOM0n = 0 (toggle mode).

(e) Timer output mode register 0 (TOM0)

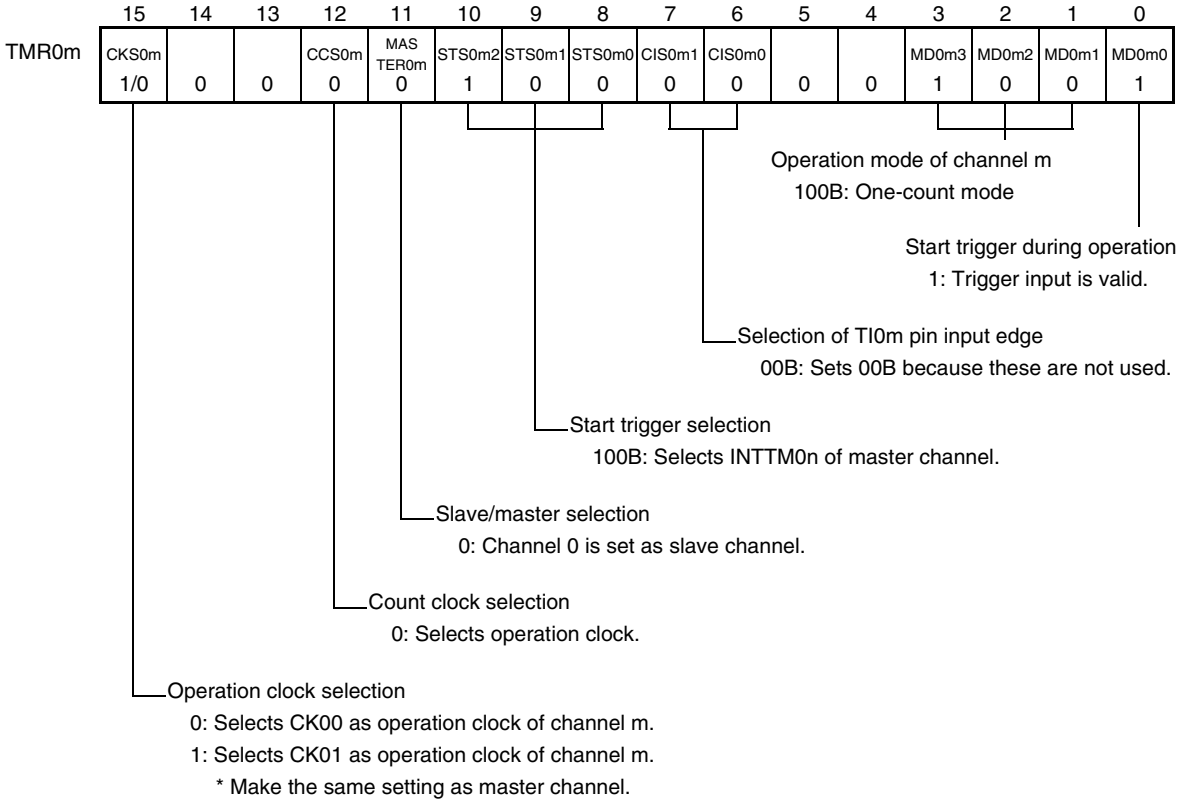
	Bit n
TOM0	TOM0n
	0

0: Sets toggle mode.

Remark n = 0, 2, 4

Figure 6-58. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register 0m (TMR0m)



(b) Timer output register 0 (TO0)

	Bit m	
TO0	TO0m 1/0	0: Outputs 0 from TO0m. 1: Outputs 1 from TO0m.

(c) Timer output enable register 0 (TOE0)

	Bit m	
TOE0	TOE0m 1/0	0: Stops the TO0m output operation by counting operation. 1: Enables the TO0m output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit m	
TOL0	TOL0m 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 1	1: Sets the combination-operation mode.

Remark n = 0, 2, 4
m = n + 1

Figure 6-59. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of 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 TMR0n and TMR0m registers of two channels to be used (determines operation mode of channels). An interval (period) value is set to the TDR0n register of the master channel, and a duty factor is set to the TDR0m register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0m bit of the TOM0 register is set to 1 (combination-operation mode). Sets the TOL0mbit. Sets the TO0m bit and determines default level of the TO0m output. →	The TO0n pin goes into Hi-Z output state. The TO0n default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOE0m to 1 and enables operation of TO0m. →	TO0m does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0m pin outputs the TO0m set level.

Remark n = 0, 2, 4
m = n + 1

Figure 6-59. Operation Procedure When PWM Function Is Used (2/2)

	Software Operation	Hardware Status
Operation start	Sets TOE0m (slave) to 1 (only when operation is resumed). The TS0n (master) and TS0m (slave) bits of the TS0 register are set to 1 at the same time. →	TE0n = 1, TE0m = 1 When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
	The TS0n and TS0m bits automatically return to 0 because they are trigger bits.	
During operation	Set values of the TMR0n and TMR0m registers cannot be changed. Set values of the TDR0n and TDR0m registers can be changed after INTTM0n of the master channel is generated. The TCR0n and TCR0m registers can always be read. The TSR0n and TSR0m registers are not used. Set values of the TOL0, TO0, and TOE0 registers cannot be changed.	The counter of the master channel loads the TDR0n value to TCR0n, and counts down. When the count value reaches TCR0n = 0000H, INTTM0n output is generated. At the same time, the value of the TDR0n register is loaded to TCR0n, and the counter starts counting down again. At the slave channel, the value of TDR0m is loaded to TCR0m, triggered by INTTM0n of the master channel, and the counter starts counting down. The output level of TO0m becomes active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0m = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n (master) and TT0m (slave) bits are set to 1 at the same time. →	TE0n, TE0m = 0, and count operation stops. TCR0n and TCR0m hold count value and stops. The TO0m output is not initialized but holds current status.
	The TT0n and TT0m bits automatically return to 0 because they are trigger bits.	
	TOE0m of slave channel is cleared to 0 and value is set to the TO0m register. →	The TO0m pin outputs the TO0n set level.
TAU stop	To hold the TO0m pin output levels Clears TO0m bit to 0 after the value to be held is set to the port register. →	The TO0m pin output levels is held by port function.
	When holding the TO0m pin output levels is not necessary Switches the port mode register to input mode. →	The TO0m pin output levels go are into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0m bit is cleared to 0 and the TO0m pin is set to port mode.)

Operation is resumed.

Remark n = 0, 2, 4
m = n + 1

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 TI0n pin.

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

Delay time = {Set value of TDR0n (master) + 2} × Count clock period
Pulse width = {Set value of TDR0m (slave)} × Count clock period

The Master channel operates in the one-count mode and counts the delays. TCR0n of the master channel starts operating upon start trigger detection and TCR0n loads the value of TDR0n. TCR0n counts down from the value of TDR0n it has loaded, in synchronization with the count clock. When TCR0n = 0000H, it outputs INTTM0n and stops counting until the next start trigger is detected.

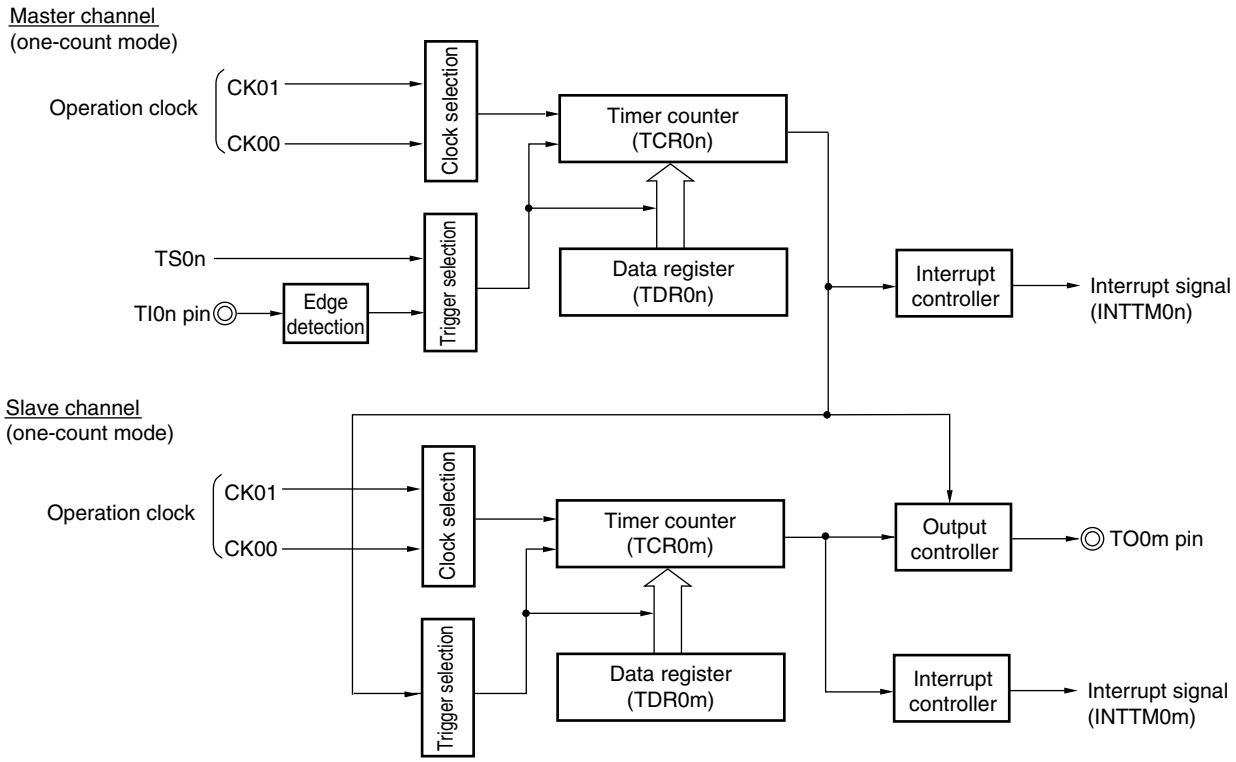
The slave channel operates in the one-count mode and counts the pulse width. TCR0m of the slave channel starts operation using INTTM0n of the master channel as a start trigger, and loads the TDR0m value. TCR0m counts down from the value of TDR0m it has loaded, in synchronization with the count value. When TCR0m = 0000H, it outputs INTTM0m and stops counting until the next start trigger (INTTM0n of the master channel) is detected. The output level of TO0m becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0m = 0000H.

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

Caution The timing of loading of TDR0n of the master channel is different from that of TDR0m of the slave channel. If TDR0n and TDR0m are rewritten during operation, therefore, an illegal waveform is output. Be sure to rewrite TDR0n and TDR0m after INTTM0n of the channel to be rewritten is generated.

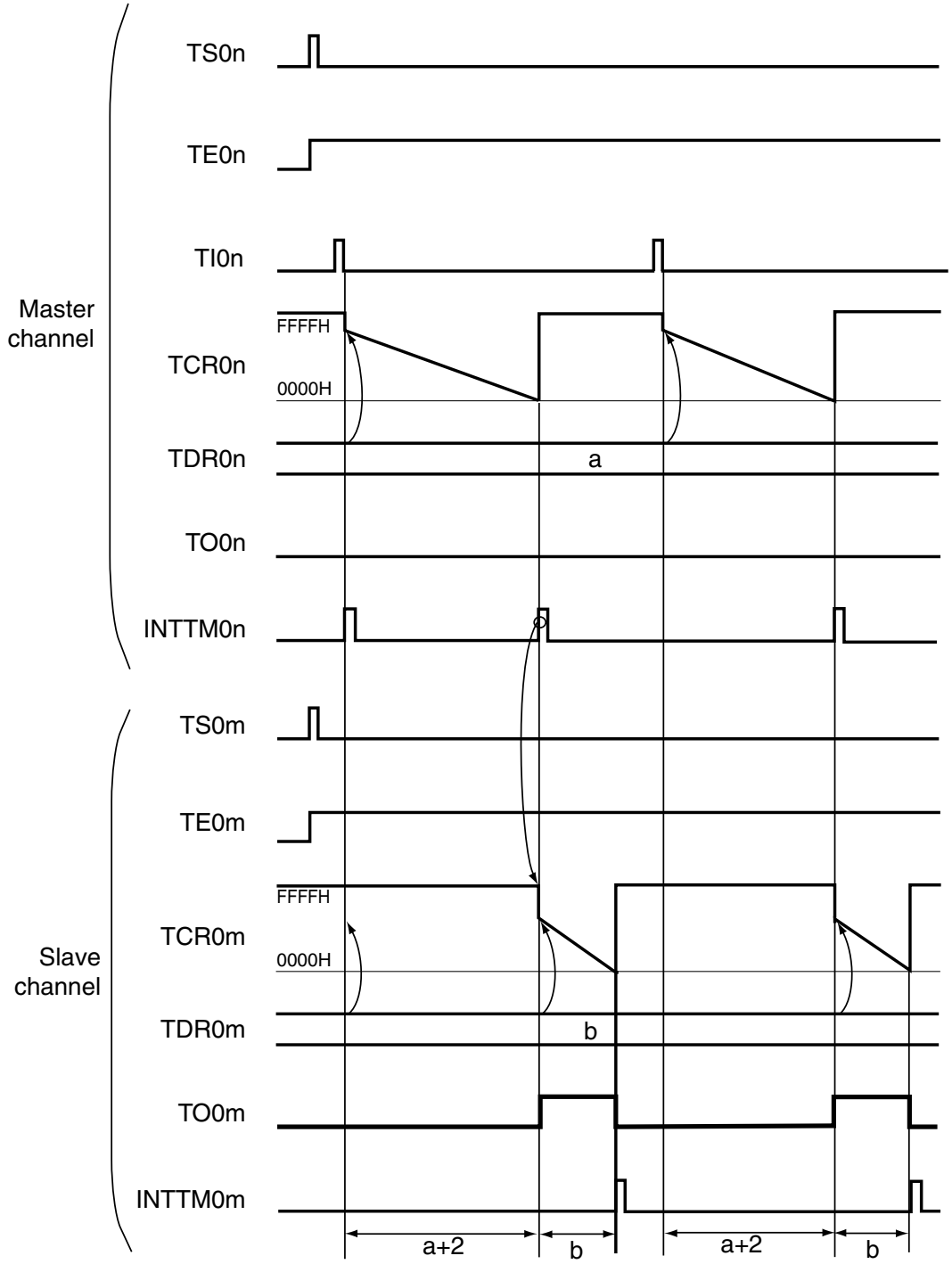
Remark n = 0, 2, 4
m = n + 1

Figure 6-60. Block Diagram of Operation as One-Shot Pulse Output Function



Remark n = 0, 2, 4
m = n + 1

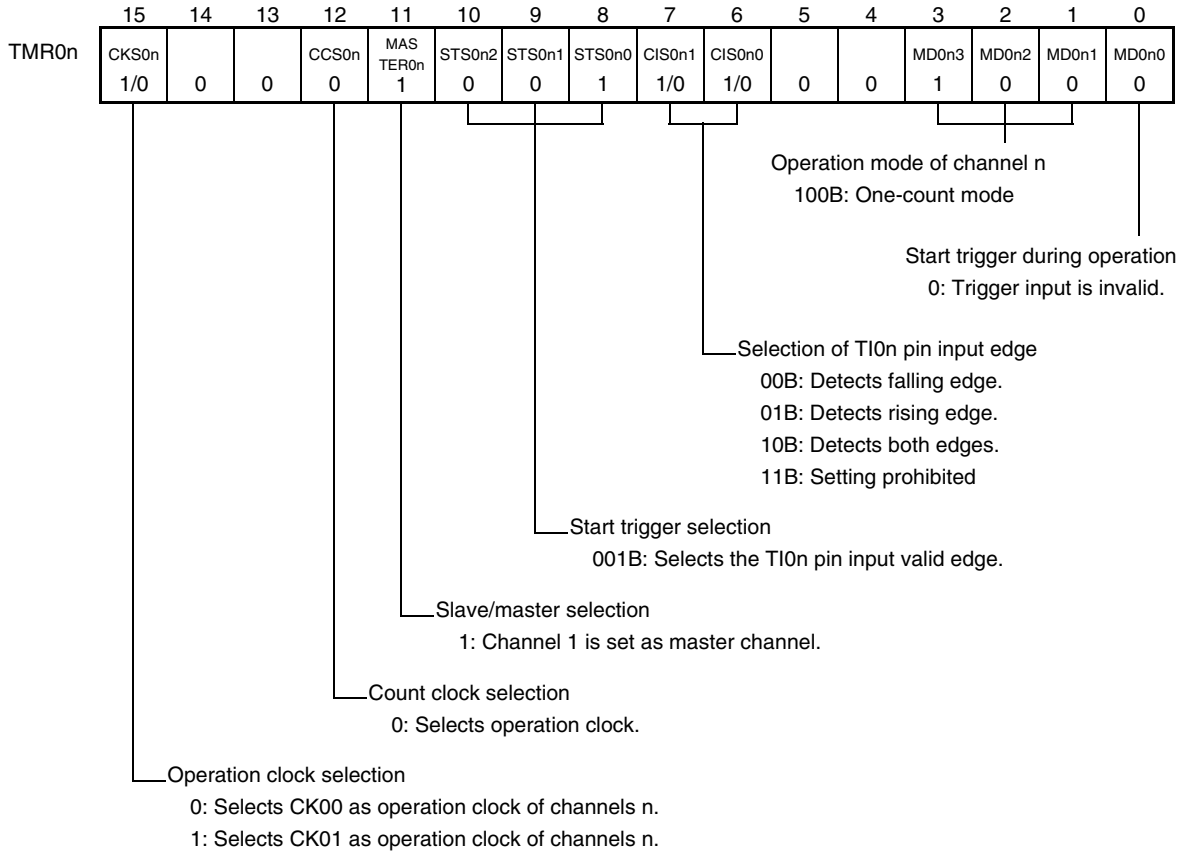
Figure 6-61. Example of Basic Timing of Operation as One-Shot Pulse Output Function



Remark n = 0, 2, 4
m = n + 1

Figure 6-62. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

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

(c) Timer output enable register 0 (TOE0)

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

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Cleared to 0 when TOM0n = 0 (toggle mode).

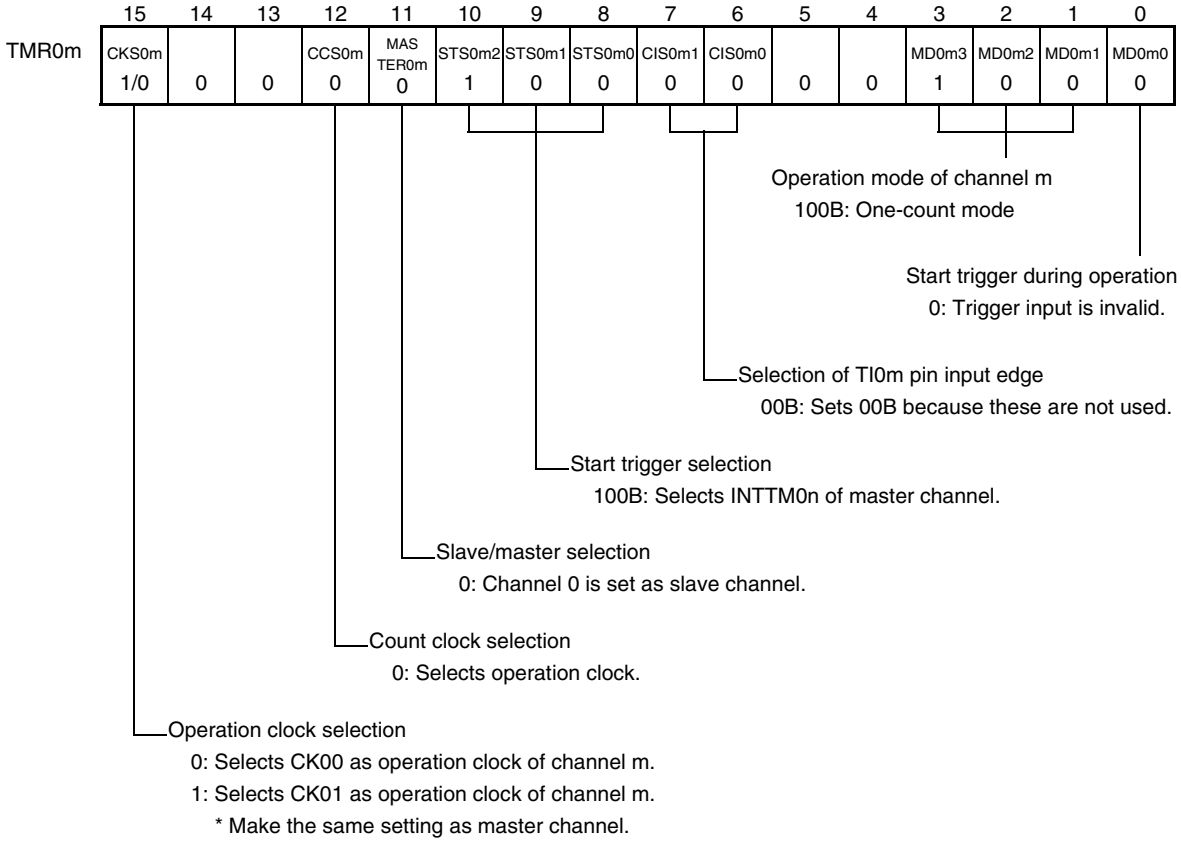
(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets toggle mode.

Remark n = 0, 2, 4

Figure 6-63. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register 0m (TMR0m)



(b) Timer output register 0 (TO0)

	Bit m	
TO0	TO0m 1/0	0: Outputs 0 from TO0m. 1: Outputs 1 from TO0m.

(c) Timer output enable register 0 (TOE0)

	Bit m	
TOE0	TOE0m 1/0	0: Stops the TO0m output operation by counting operation. 1: Enables the TO0m output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit m	
TOL0	TOL0m 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 1	1: Sets the combination-operation mode.

Remark n = 0, 2, 4
m = n + 1

Figure 6-64. Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of 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 TMR0n and TMR0m registers of two channels to be used (determines operation mode of channels). An output delay is set to the TDR0n register of the master channel, and a pulse width is set to the TDR0m register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0m bit of the TOM0 register is set to 1 (combination-operation mode). Sets the TOL0m bit. Sets the TO0m bit and determines default level of the TO0m output. →	The TO0n pin goes into Hi-Z output state. The TO0n default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOE0m to 1 and enables operation of TO0m. →	TO0m does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0m pin outputs the TO0m set level.

Remark n = 0, 2, 4
m = n + 1

Figure 6-64. Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
Operation start	Sets TOE0m (slave) to 1 (only when operation is resumed). The TS0n (master) and TS0m (slave) bits of the TS0 register are set to 1 at the same time. →	TE0n and TE0m are set to 1 and the master channel enters the TI0n input edge detection wait status. Counter stops operating.
	Detects the TI0n pin input valid edge of master channel. →	Master channel starts counting.
During operation	Set values of only the CISn1 and CISn0 bits of the TMR0n register can be changed. Set values of the TMR0m, TDR0n, TDR0m, and TOM0 registers cannot be changed. The TCR0n and TCR0m registers can always be read. The TSR0n and TSR0m registers are not used. Set values of the TOL0, TO0, and TOE0 registers can be changed.	Master channel loads the value of TDR0n to TCR0n when the TI0n pin valid input edge is detected, and the counter starts counting down. When the count value reaches TCR0n = 0000H, the INTTM0n output is generated, and the counter stops until the next valid edge is input to the TI0n pin. The slave channel, triggered by INTTM0n of the master channel, loads the value of TDR0m to TCR0m, and the counter starts counting down. The output level of TO0m becomes active one count clock after generation of INTTM0n from the master channel. It becomes inactive when TCR0m = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n (master) and TT0m (slave) bits are set to 1 at the same time. →	TE0n, TE0m = 0, and count operation stops. TCR0n and TCR0m hold count value and stops. The TO0m output is not initialized but holds current status.
	TOE0m of slave channel is cleared to 0 and value is set to the TO0 register. →	The TO0m pin outputs the TO0n set level.
TAU stop	To hold the TO0m pin output levels Clears TO0m bit to 0 after the value to be held is set to the port register. →	The TO0m pin output levels is held by port function.
	When holding the TO0m pin output levels is not necessary Switches the port mode register to input mode. →	The TO0m pin output levels go are into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0m bit is cleared to 0 and the TO0m pin is set to port mode.)

Operation is resumed.

Remark n = 0, 2, 4
m = n + 1

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 TDR0n (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDR0m (slave 1)}/{Set value of TDR0n (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDR0m (slave 2)}/{Set value of TDR0n (master) + 1} × 100

Remark Although the duty factor exceeds 100% if the set value of TDR0p (slave 1) > {set value of TDR0n (master) + 1} or if the {set value of TDR0q (slave 2)} > {set value of TDR0n (master) + 1}, it is summarized into 100% output.

TCR0n of the master channel operates in the interval timer mode and counts the periods.

TCR0p of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0p pin. TCR0p loads the value of TDR0p to TCR0p, using INTTM0n of the master channel as a start trigger, and start counting down. When TCR0p = 0000H, TCR0p outputs INTTM0p and stops counting until the next start trigger (INTTM0n of the master channel) has been input. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H.

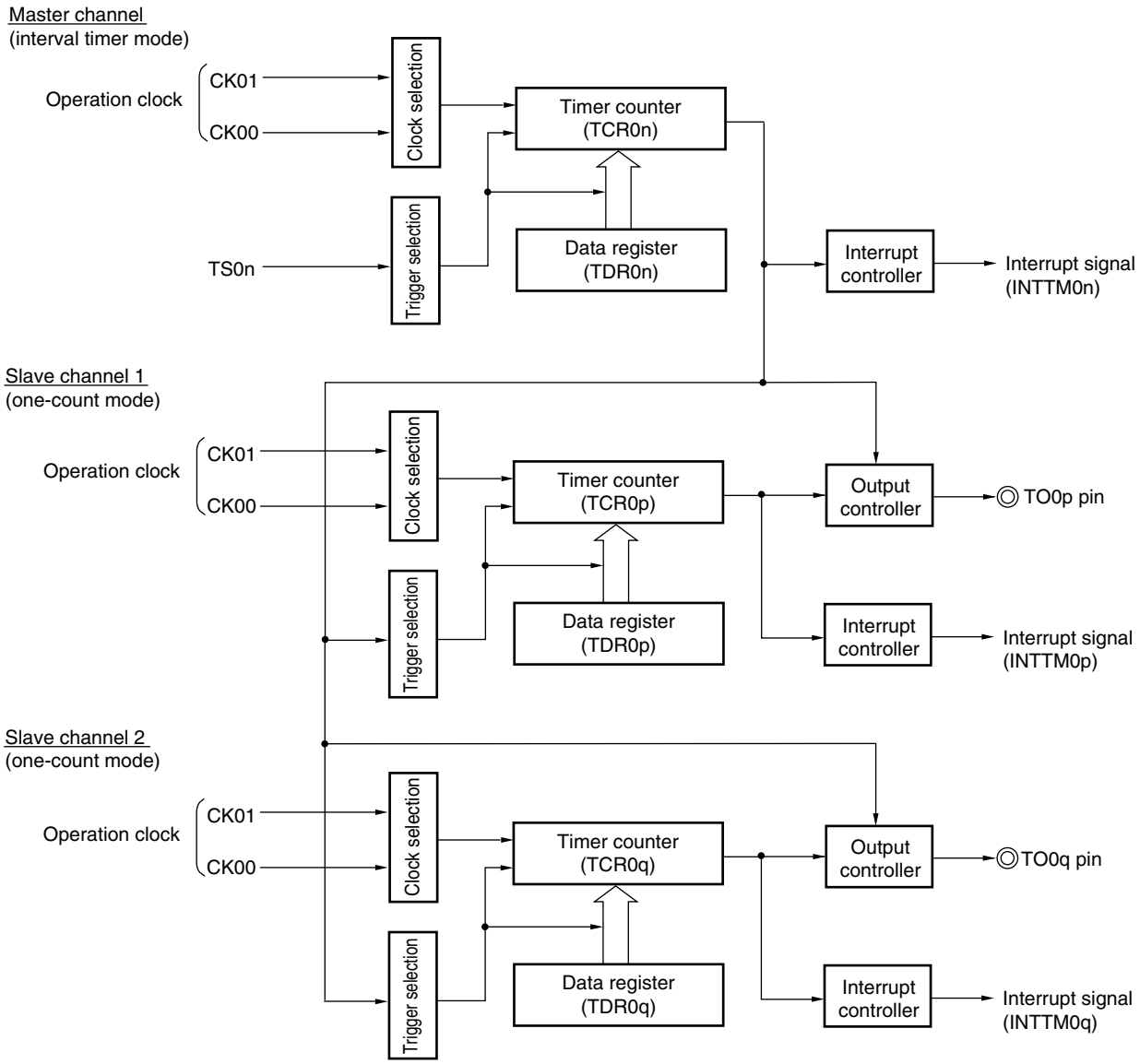
In the same way as TCR0p of the slave channel 1, TCR0q of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0q pin. TCR0q loads the value of TDR0q to TCR0q, using INTTM0n of the master channel as a start trigger, and starts counting down. When TCR0q = 0000H, TCR0q outputs INTTM0q and stops counting until the next start trigger (INTTM0n of the master channel) has been input. The output level of TO0q becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0q = 0000H.

When channel 0 is used as the master channel as above, up to seven types of PWM signals can be output at the same time.

Caution To rewrite both TDR0n of the master channel and TDR0p of the slave channel 1, write access is necessary at least twice. Since the values of TDR0n and TDR0p are loaded to TCR0n and TCR0p after INTTM0n is generated from the master channel, if rewriting is performed separately before and after generation of INTTM0n from the master channel, the TO0p pin cannot output the expected waveform. To rewrite both TDR0n of the master and TDR0p of the slave, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel (This applies also to TDR0q of the slave channel 2) .

Remarks 1. n = 0, 2, 4
n < p < q ≤ 6
Where p and q are consecutive integers following n (p = n + 1, q = n + 2)

Figure 6-65. Block Diagram of Operation as Multiple PWM Output Function (output two types of PWMs)

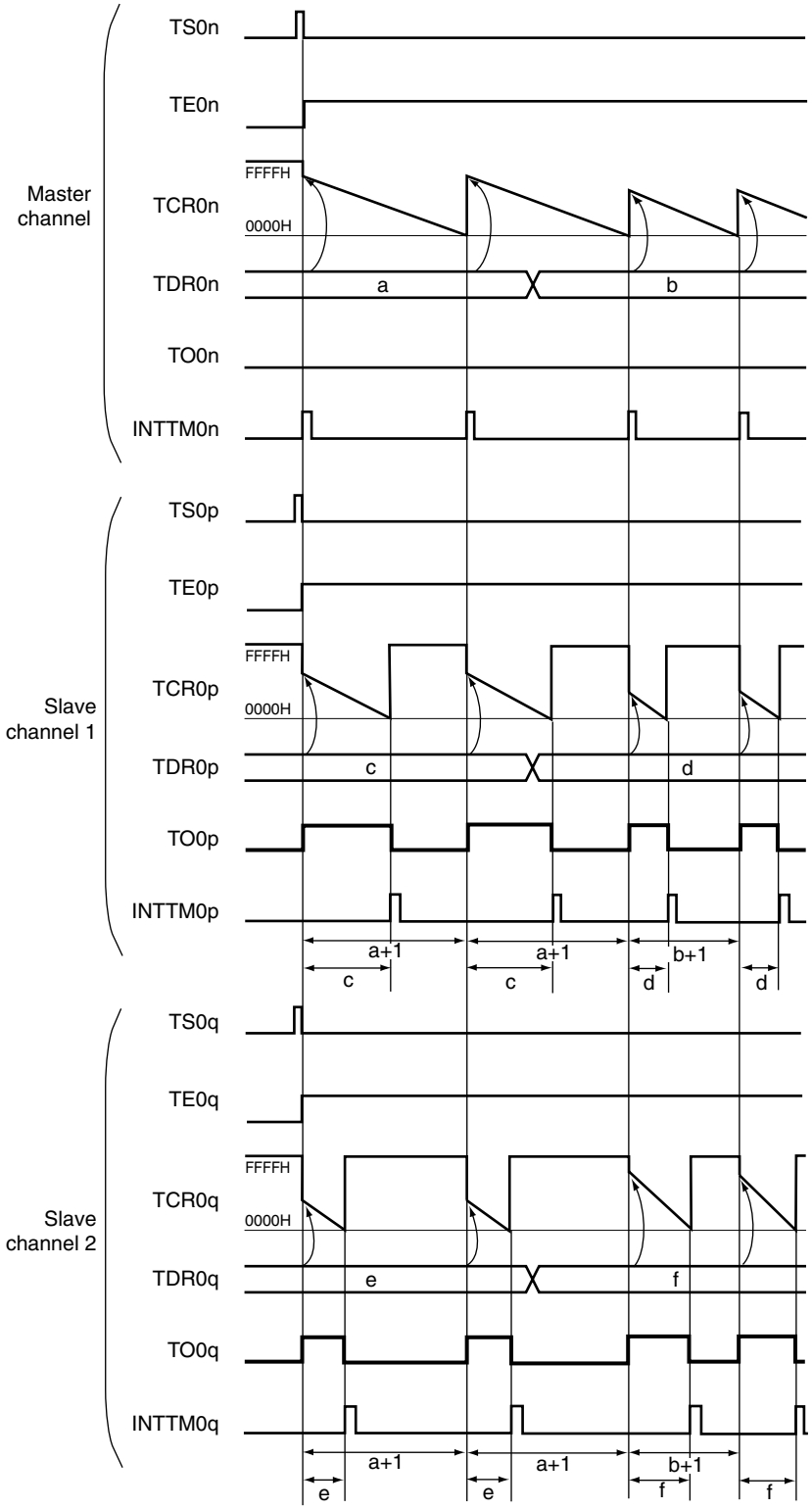


Remarks 1. n = 0, 2, 4

2. p = n + 1

q = n + 2

Figure 6-66. Example of Basic Timing of Operation as Multiple PWM Output Function (output two types of PWMs)



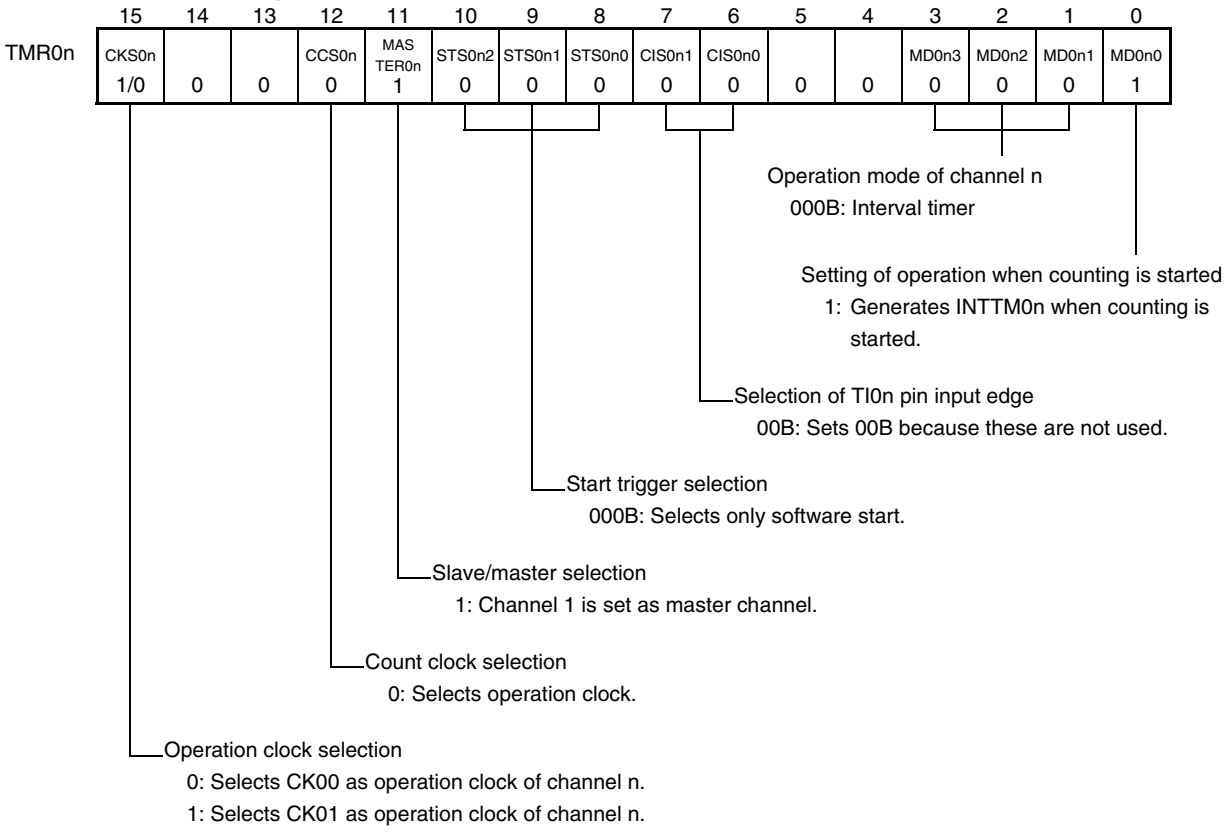
Remarks 1. n = 0, 2, 4

2. p = n + 1

q = n + 2

Figure 6-67. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

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

(c) Timer output enable register 0 (TOE0)

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

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n	0: Cleared to 0 when TOM0n = 0 (toggle mode).
	0	

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n	0: Sets toggle mode.
	0	

Remark n = 0, 2, 4

Figure 6-68. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)

(a) Timer mode register 0p, 0q (TMR0p, TMR0q)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0p	CKS0p			CCS0p	MAS TER0p	STS0p2	STS0p1	STS0p0	CIS0p1	CIS0p0			MD0p3	MD0p2	MD0p1	MD0p0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1
TMR0q	CKS0q			CCS0q	MAS TER0q	STS0q2	STS0q1	STS0q0	CIS0q1	CIS0q0			MD0q3	MD0q2	MD0q1	MD0q0
	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 TI0p and TI0q pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTM0n of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CK00 as operation clock of channel p, q.
1: Selects CK01 as operation clock of channel p, q.
* Make the same setting as master channel.

(b) Timer output register 0 (TO0)

	Bit q	Bit p	
TO0	TO0q	TO0p	0: Outputs 0 from TO0p or TO0q. 1: Outputs 1 from TO0p or TO0q.
	1/0	1/0	

(c) Timer output enable register 0 (TOE0)

	Bit q	Bit p	
TOE0	TOE0q	TOE0p	0: Stops the TO0p or TO0q output operation by counting operation. 1: Enables the TO0p or TO0q output operation by counting operation.
	1/0	1/0	

(d) Timer output level register 0 (TOL0)

	Bit q	Bit p	
TOL0	TOL0q	TOL0p	0: Positive logic output (active-high) 1: Inverted output (active-low)
	1/0	1/0	

(e) Timer output mode register 0 (TOM0)

	Bit q	Bit p	
TOM0	TOM0q	TOM0p	1: Sets the combination-operation mode.
	1	1	

Remark n = 0, 2, 4; p = n+1; q = n+2

Figure 6-69. Operation Procedure When Multiple PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of 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 TMR0n, TMR0p, and TMR0q registers of each channel to be used (determines operation mode of channels). An interval (period) value is set to the TDR0n register of the master channel, and a duty factor is set to the TDR0m register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0m bit of the TOM0 register is set to 1 (combination-operation mode). Clears the TOL0p and TOL0q bits to 0. Sets the TO0p and TO0q bits and determines default level of the TO0p and TO0q outputs. →	The TO0n pin goes into Hi-Z output state. The TO0p and TO0q default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets TOE0p or TOE0q to 1 and enables operation of TO0m. →	TO0p or TO0q does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0p and TO0q pins output the TO0p and TO0q set levels.

Remarks 1. n = 0, 2, 4

2. p = n + 1; q = n + 2

Figure 6-69. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets TOE0p and TOE0q (slave) to 1 (only when operation is resumed). The TS0n bit (master), and TS0p and TS0q (slave) bits of the TS0 register are set to 1 at the same time. →	TE0n = 1, TE0p, TE0q = 1 When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
	The TS0n, TS0p, and TS0q bits automatically return to 0 because they are trigger bits.	
During operation	Set values of the TMR0n, TMR0p, TMR0q, TOM0, and TOE0 registers cannot be changed. Set values of the TDR0n, TDR0p, and TDR0q registers can be changed after INTTM0n of the master channel is generated. The TCR0n, TCR0p, and TCR0q registers can always be read. The TSR0n, TSR0p, and TSR0q registers are not used. Set values of the TOM0, TOL0, TO0, and TOE0 registers can be changed.	The counter of the master channel loads the TDR0n value to TCR0n and counts down. When the count value reaches TCRn = 0000H, INTTM0n output is generated. At the same time, the value of the TDR0n register is loaded to TCR0n, and the counter starts counting down again. At the slave channel 1, the values of TDR0p are transferred to TCR0p, triggered by INTTM0n of the master channel, and the counter starts counting down. The output levels of TO0p become active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. At the slave channel 2, the values of TDR0q are transferred to TDR0q, triggered by INTTM0n of the master channel, and the counter starts counting down. The output levels of TO0q become active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0q = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n bit (master), TT0p, and TT0q (slave) bits are set to 1 at the same time. →	TE0n, TE0p, TE0q = 0, and count operation stops. TCR0n, TCR0p, and TCR0q hold count value and stops. The TO0m output is not initialized but holds current status.
	The TT0n, TT0p, and TT0q bits automatically return to 0 because they are trigger bits.	
	TOE0p or TOE0q of slave channel is cleared to 0 and value is set to the TO0p and TO0q registers. →	The TO0p and TO0q pins output the TO0p and TO0q set levels.
TAU stop	To hold the TO0p and TO0q pin output levels Clears TO0p and TO0q bits to 0 after the value to be held is set to the port register. →	The TO0p and TO0q pin output levels are held by port function.
	When holding the TO0p and TO0q pin output levels is not necessary Switches the port mode register to input mode. →	The TO0p and TO0q pin output levels go into Hi-Z output state.
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0p and TO0q bits are cleared to 0 and the TO0p and TO0q pins are set to port mode.)

Remarks 1. n = 0, 2, 4

2. p = n + 1; q = n + 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

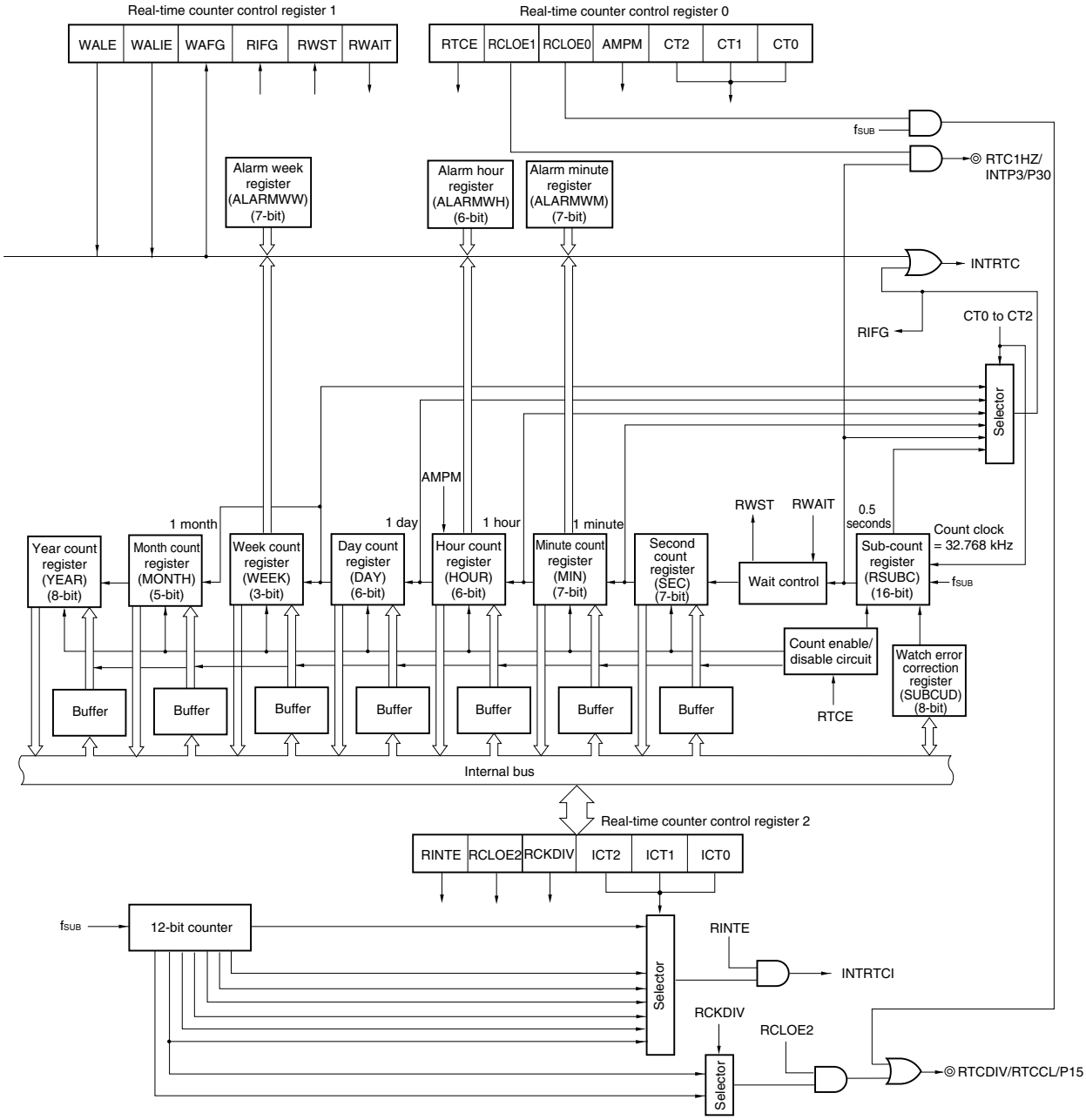
7.2 Configuration of Real-Time Counter

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)

Figure 7-1. Block Diagram of Real-Time Counter



7.3 Registers Controlling Real-Time Counter

Timer real-time counter is controlled by the following 16 registers.

(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	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN

RTCEN	Control of real-time counter (RTC) input clock[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 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.

<R> **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. Be sure to clear bits 1 and 6 of 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 kHz).
1	Enables output of RTCCL pin (32 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, and re-set the hour count register (HOUR). • Table 7-2 shows the displayed time digits that are displayed.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
After changing the values of CT2 to CT0, clear the interrupt request flag.			

Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Caution If RCLOE0 and RCLOE1 are changed when RTCE = 1, a pulse with a narrow width may be generated on the 32 kHz and 1 kHz output signals.

Remark ×: don't care

Table 7-2. Displayed Time Digits

24-Hour System	12-Hour System	24-Hour System	12-Hour System
00	12 (AM12)	12	32 (PM12)
01	01 (AM1)	13	21 (PM1)
02	02 (AM2)	14	22 (PM2)
03	03 (AM3)	15	23 (PM3)
04	04 (AM4)	16	24 (PM4)
05	05 (AM5)	17	25 (PM5)
06	06 (AM6)	18	26 (PM6)
07	07 (AM7)	19	27 (PM7)
08	08 (AM8)	20	28 (PM8)
09	09 (AM9)	21	29 (PM9)
10	10 (AM10)	22	30 (PM10)
11	11 (AM11)	23	31 (PM11)

(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.
To set the registers of alarm (WALIE flag of RTCC1, ALARMWM register, ALARMWH register, and ALARMWW register), disable WALE (clear it to “0”).	

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 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 kHz) until the counter value can be read or written. If RSUBC overflows when RWAIT = 1, it counts up after RWAIT = 0. If the second count register is written, however, it does not count up because RSUBC is cleared.	

Caution If writing is performed to the WAFG flag with a 1-bit manipulation instruction, the RIFG flag may be cleared. Therefore, to perform writing to the WAFG flag, be sure to use an 8-bit manipulation instruction, and at this time, set 1 to the RIFG flag to invalidate writing. In the same way, to perform writing to the RIFG flag, use an 8-bit manipulation instruction and set 1 the WAFR flag.

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)
Change ICT2, ICT1, and ICT0 when RINTE = 0.				

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	RTCDIV pin outputs 16.384 kHz.

Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Caution 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 pluse of at least one clock width of f_{XT} may be generated.

(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 this 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. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside this 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 0 to 23 or 1 to 12 (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. 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 this 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).

(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. Set a decimal value of 00 to 31 to this register in BCD code. If a value outside this 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 this 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

(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. Set a decimal value of 01 to 12 to this register in BCD code. If a value outside this 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. Set a decimal value of 00 to 99 to this register in BCD code. If a value outside this 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 count value of the sub-count register (RSUBC).

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.
1	Corrects watch error only when the second digits are at 00.

F6	Setting of watch error correction method
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. /F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100).	

(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.

<R> However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 7-16. 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.

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 7-17. 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

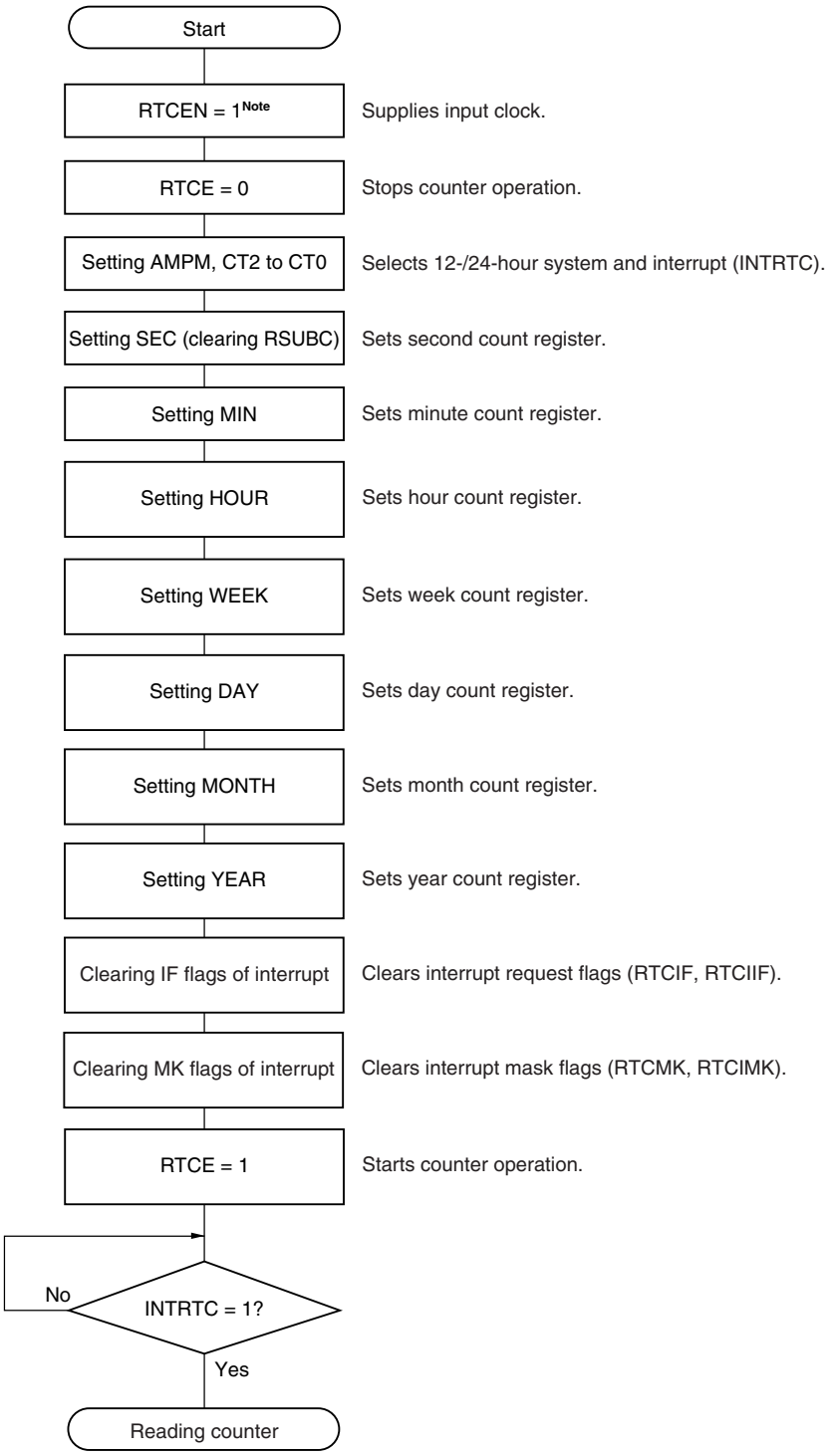
Here is an example of setting the alarm.

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

7.4 Real-Time Counter Operation

7.4.1 Starting operation of real-time counter

Figure 7-18. Procedure for Starting Operation of Real-Time Counter

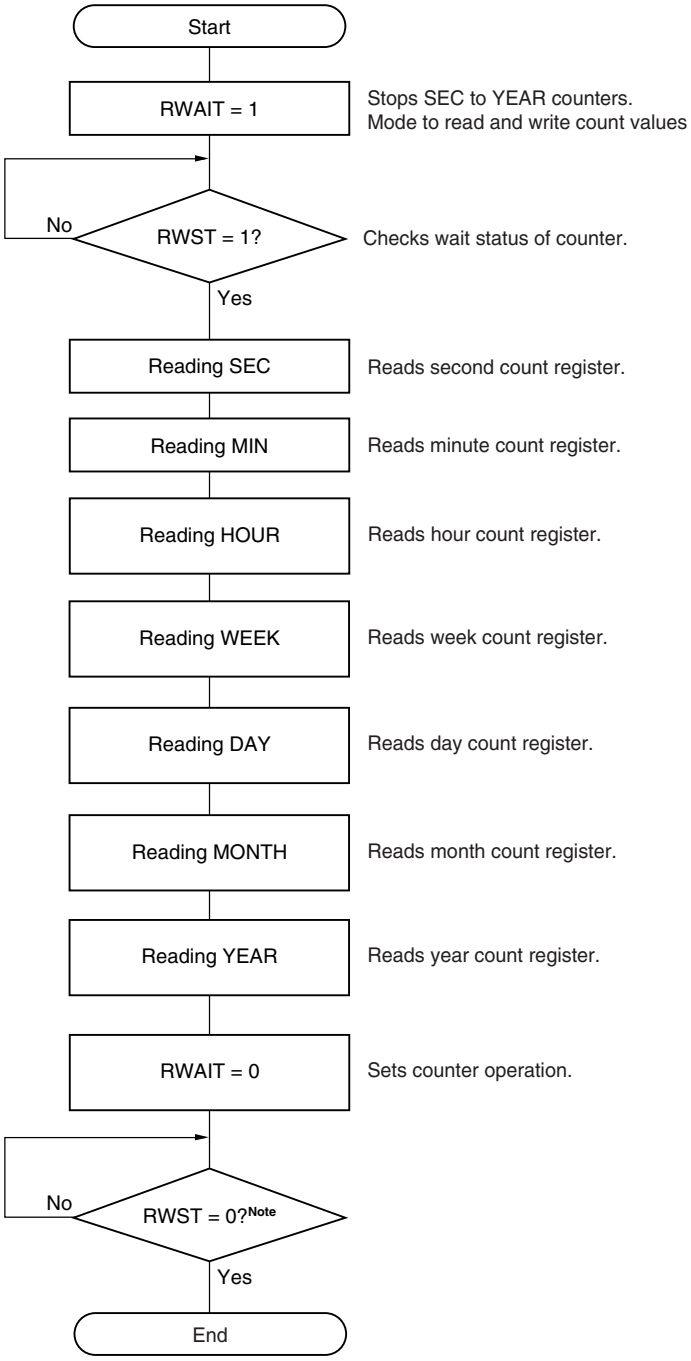


<R> **Note** First set RTCEN to 1, while oscillation of the subsystem clock (f_{SUB}) is stable.

7.4.2 Reading/writing real-time counter

Read or write the counter when RWAIT = 1.

Figure 7-19. Procedure for Reading Real-Time Counter

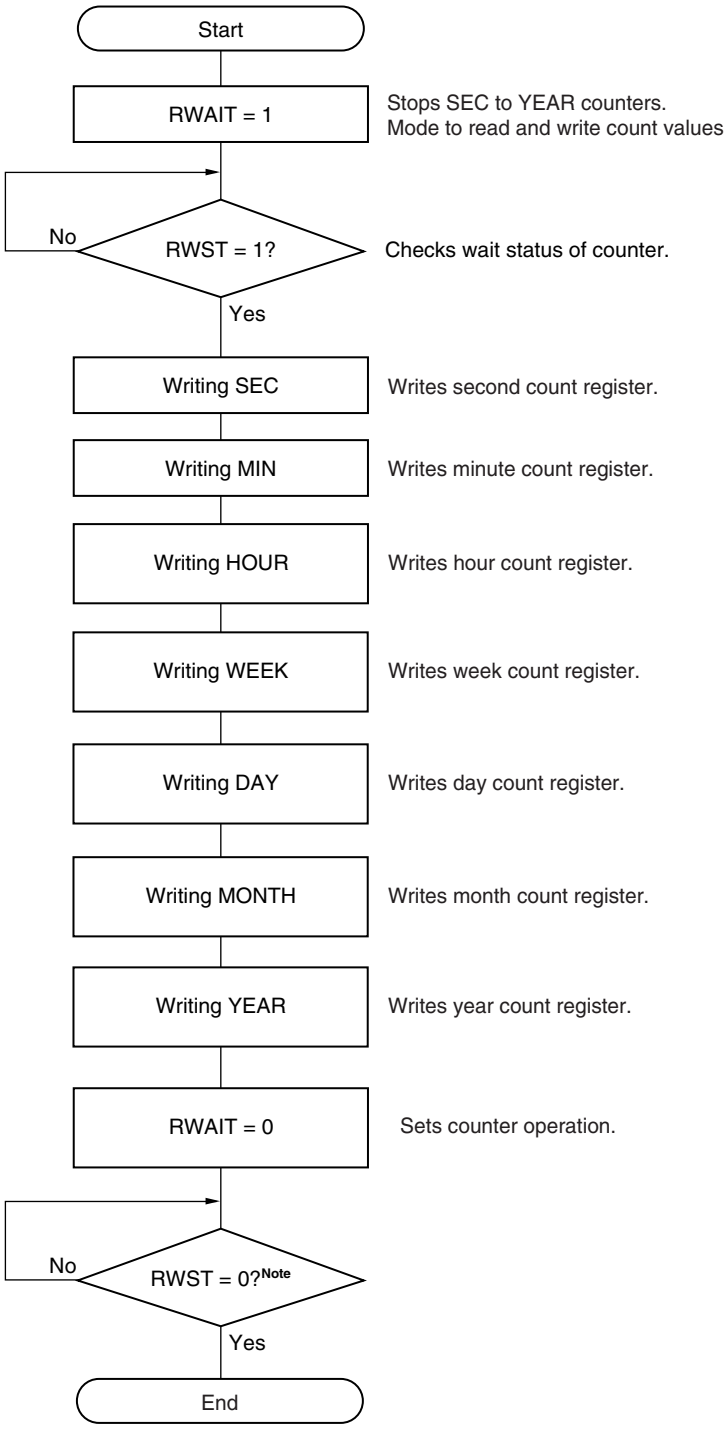


Note Be sure to confirm that RWST = 0 before setting STOP mode.

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 7-20. Procedure for Writing Real-Time Counter



Note Be sure to confirm that RWST = 0 before setting STOP mode.

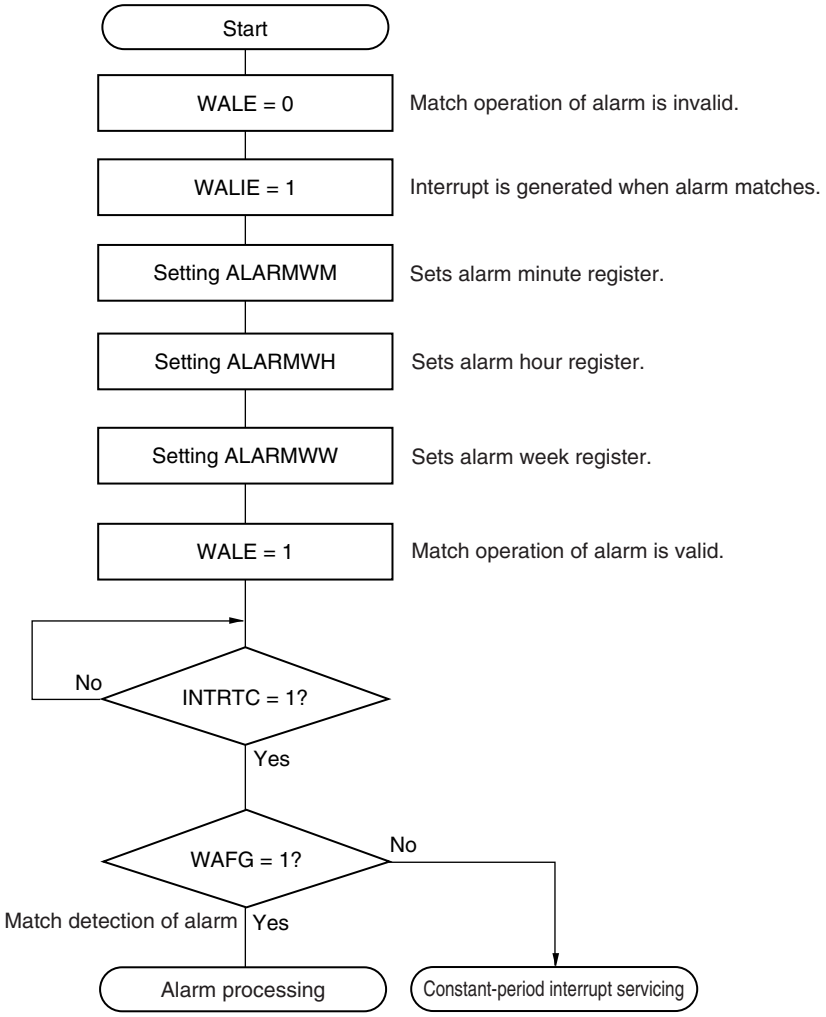
Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

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.

7.4.3 Setting alarm of real-time counter

Set time of alarm when WALE = 0.

Figure 7-21. 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.

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 18 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)

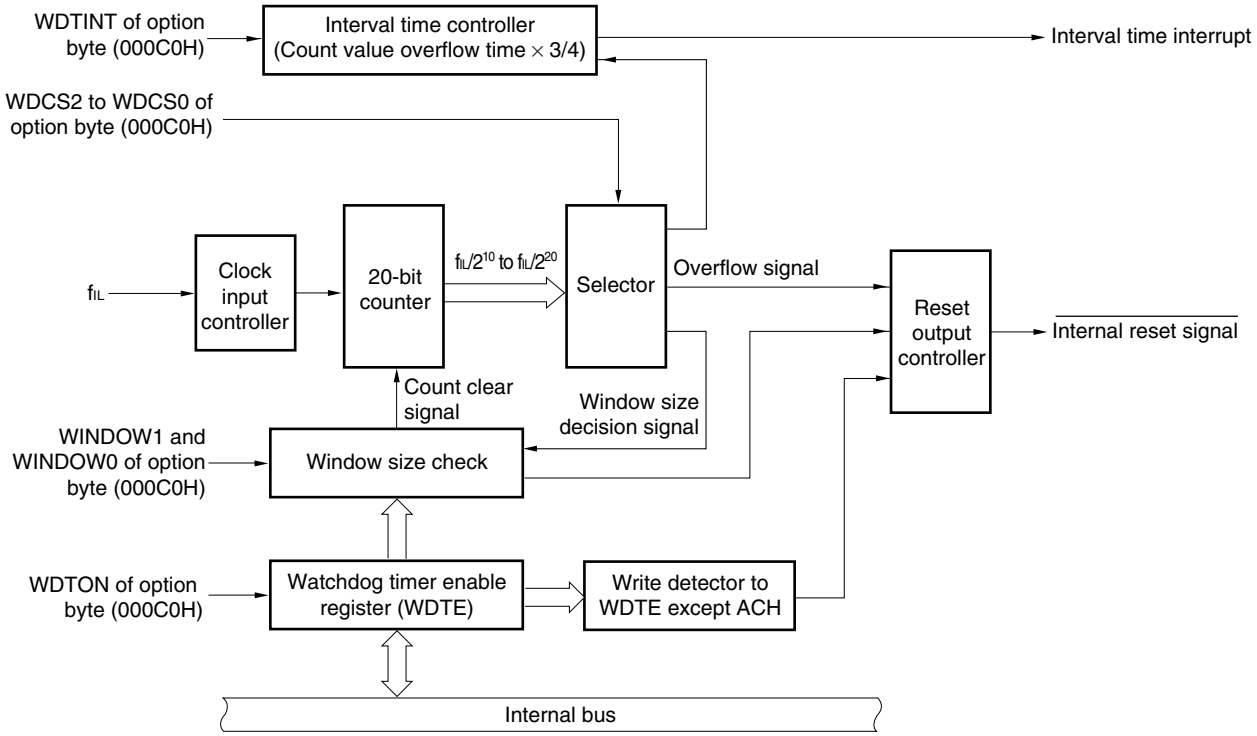
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

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 22 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

<R> **Cautions 1. If a value other than “ACH” is written to WDTE, an internal reset signal is generated.**

<R> **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 22**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **8.4.2** and **CHAPTER 22**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **8.4.3** and **CHAPTER 22**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to WDTE after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write WDTE the second time or later after a reset release during the window open period. If WDTE is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to WDTE, an internal reset signal is generated.
   A internal reset signal is generated in the following cases.

   - If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
   - If data other than “ACH” is written to WDTE

Cautions 1. When data is written to WDTE for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

2. If the watchdog timer is cleared by writing “ACH” to WDTE, the actual overflow time may be different from the overflow time set by the option byte by up to $2/f_{IL}$ seconds.

3. The watchdog timer can be cleared immediately before the count value overflows.

<Example> When the overflow time is set to $2^{10}/f_{IL}$, writing “ACH” is valid up to count value 3FH.

Cautions 4. The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		

If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts.
When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

5. The watchdog timer continues its operation during self-programming of the flash memory and EEPROM™ emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

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 time is set.

Table 8-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer
0	0	0	$2^{10}/f_{IL}$ (3.88 ms)
0	0	1	$2^{11}/f_{IL}$ (7.76 ms)
0	1	0	$2^{12}/f_{IL}$ (15.52 ms)
0	1	1	$2^{13}/f_{IL}$ (31.03 ms)
1	0	0	$2^{15}/f_{IL}$ (124.12 ms)
1	0	1	$2^{17}/f_{IL}$ (496.48 ms)
1	1	0	$2^{18}/f_{IL}$ (992.97 ms)
1	1	1	$2^{20}/f_{IL}$ (3971.88 ms)

Caution The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

Remarks 1. f_{IL}: Internal low-speed oscillation clock frequency

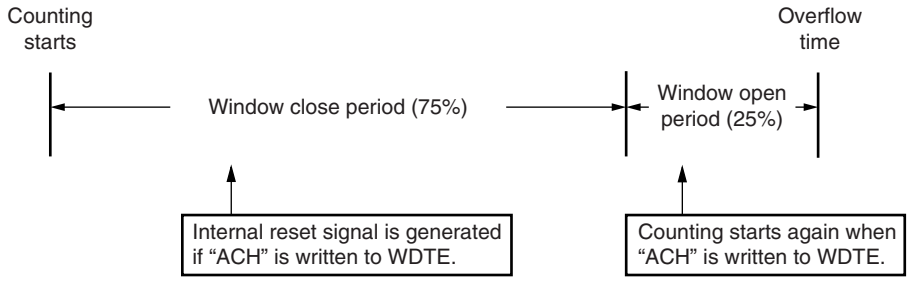
2. ( ): f_{IL} = 264 kHz (MAX.)

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 to 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 used at a supply voltage (V_{DD}) below 2.7 V.**
- **When stopping all main system clocks (internal high-speed oscillation clock, X1 clock, and external main system clock) by use of the STOP mode or software.**

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 3.56 ms	0 to 2.37 ms	0 to 0.119 ms	None
Window open time	3.56 to 3.88 ms	2.37 to 3.88 ms	0.119 to 3.88 ms	0 to 3.88 ms

<When window open period is 25%>

- Overflow time:
  $2^{10}/f_{IL}$ (MAX.) = 2^{10}/264 kHz (MAX.) = 3.88 ms
- Window close time:
  0 to $2^{10}/f_{IL}$ (MIN.) × (1 − 0.25) = 0 to 2^{10}/216 kHz (MIN.) × 0.75 = 0 to 3.56 ms
- Window open time:
  $2^{10}/f_{IL}$ (MIN.) × (1 − 0.25) to $2^{10}/f_{IL}$ (MAX.) = 2^{10}/216 kHz (MIN.) × 0.75 to 2^{10}/264 kHz (MAX.)
  = 3.56 to 3.88 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.

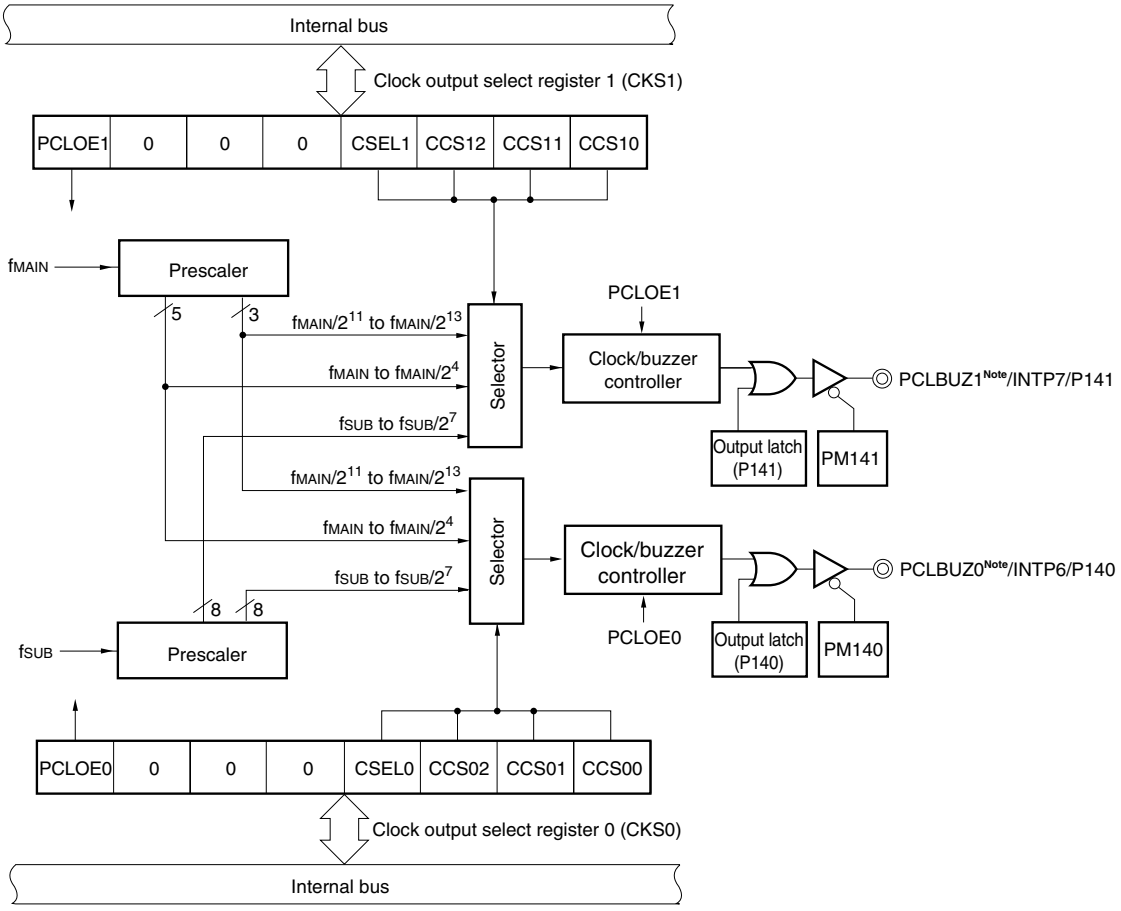
Two output pins, PCLBUZ0 and PCLBUZ1, are available.

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}. Setting a clock exceeding 5 MHz at V_{DD} < 2.7 V is prohibited.

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	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 14 (PM14)

(1) Clock output select registers 0, 1 (CKS0, CKS1)

These registers set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZ0/PCLBUZ1), and set the output clock.

Select the clock to be output from PCLBUZ0 by using CKS0.

Select the clock to be output from PCLBUZ1 by using CKS1.

CKS0 and CKS1 are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 9-2. Format of Clock Output Select Register n (CKSn)

Address: FFFA5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn output enable/disable specification
0	Output disable (default)
1	Output enable

CSELn	CCSn2	CCSn1	CCSn0	PCLBUZn output clock selection			
					f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz
0	0	0	0	f_{MAIN}	5 MHz	10 MHz[Note]	Setting prohibited[Note]
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz[Note]
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 when 2.7 V ≤ V_{DD}. Setting a clock exceeding 5 MHz at V_{DD} < 2.7 V is also prohibited.

Cautions 1. Change the output clock after disabling clock output (PCLOEn = 0).

2. If the selected clock (f_{MAIN} or f_{SUB}) stops during clock output (PCLOEn = 1), the output becomes undefined.

Remarks 1. n = 0, 1

2. f_{MAIN}: Main system clock frequency

3. f_{SUB}: Subsystem clock frequency

(2) Port mode register 14 (PM14)

This register sets port 14 input/output in 1-bit units.

When using the P140/INTP6/PCLBUZ0 and P141/INTP7/PCLBUZ1 pins for clock output/buzzer output, clear PM140 and PM141 and the output latches of P140 and P141 to 0.

PM14 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 9-3. Format of Port Mode Register 14 (PM14)

Address: FFF2EH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM14	1	1	1	1	1	1	PM141	PM140

PM14n	P14n pin I/O mode selection (n = 0, 1)
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

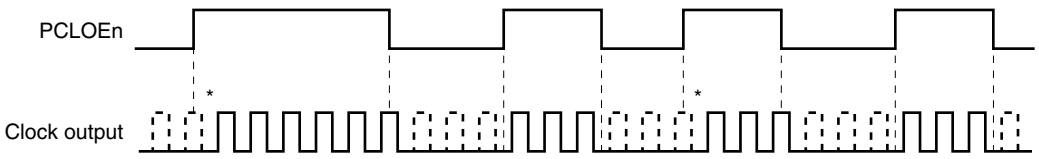
PCLBUZn is output as the following procedure.

<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.

Remark The controller is designed not to output a pulse with a narrow width when it is used to output a clock and when clock output is enabled or disabled. As shown in Figure 9-4, be sure to start output from the low period of the clock (marked with * in the figure). When stopping output, do so after the high-level period of the clock.

Figure 9-4. Remote Control Output Application Example



Remark n = 0, 1

CHAPTER 10 A/D CONVERTER

10.1 Function of A/D Converter

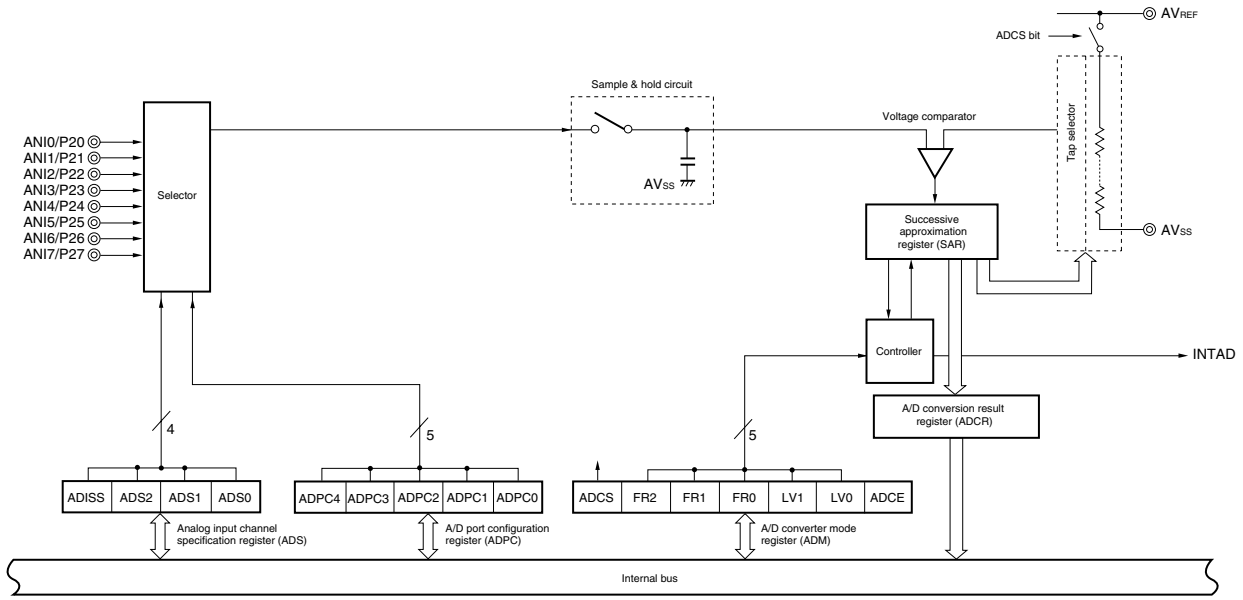
The A/D converter converts an analog input signal into a digital value, and consists of up to 8 channels (ANI0 to ANI7) with a resolution of 10 bits.

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 ANI7. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

Figure 10-1. Block Diagram of A/D Converter



10.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI7 pins

These are the analog input pins of the 8-channel A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

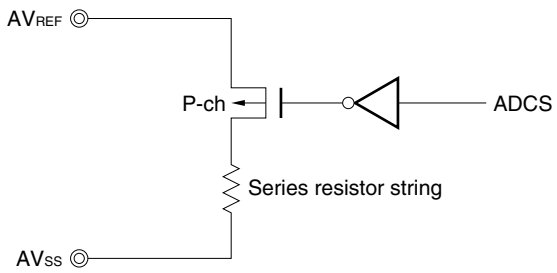
(2) Sample & hold circuit

The sample & hold circuit samples the input voltage of the analog input pin selected by the selector when A/D conversion is started, and holds the sampled voltage value during A/D conversion.

(3) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS}, and generates a voltage to be compared with the sampled voltage value.

Figure 10-2. Circuit Configuration of Series Resistor String



(4) Voltage comparator

The voltage comparator compares the sampled voltage value and the output voltage of the series resistor string.

(5) Successive approximation register (SAR)

This register converts the result of comparison by the voltage comparator, starting from the most significant bit (MSB).

When the voltage value is converted into a digital value down to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred to the A/D conversion result register (ADCR).

(6) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD.

(9) AV_{REF} pin

This pin inputs an analog power/reference voltage to the A/D converter. When all pins of port 2 are used as the analog port pins, make the potential of AV_{REF} be such that $2.3\ V \leq AV_{REF} \leq V_{DD}$. When one or more of the pins of port 2 are used as the digital port pins or when the A/D converter is not used, make AV_{REF} the same potential as EV_{DD} or V_{DD}.

The signal input to ANI0 to ANI7 is converted into a digital signal, based on the voltage applied across AV_{REF} and AV_{SS}.

(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 EV_{SS} and V_{SS} pins even when the A/D converter is not used.

(11) A/D converter mode register (ADM)

This register is used to set the conversion time of the analog input signal to be converted, and to start or stop the conversion operation.

(12) A/D port configuration register (ADPC)

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

(13) Analog input channel specification register (ADS)

This register is used to specify the port that inputs the analog voltage to be converted into a digital signal.

(14) Port mode registers 2 (PM2)

This register switches the ANI0/P20 to ANI7/P27 pins to input or output.

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 (PM2)
- 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.

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 bits 1 and 6 of PER0 register to 0.

Figure 10-3. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
PER0	RTCEN	0	ADCEN	IIC0EN	SAU1EN	SAU0EN	0	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.

(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 10-4. Format of A/D Converter Mode Register (ADM)

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	0	FR2[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

ADCE	Comparator operation control[Note 2]
0	Stops comparator operation
1	Enables comparator operation (comparator: 1/2AV$_{REF}$ operation)

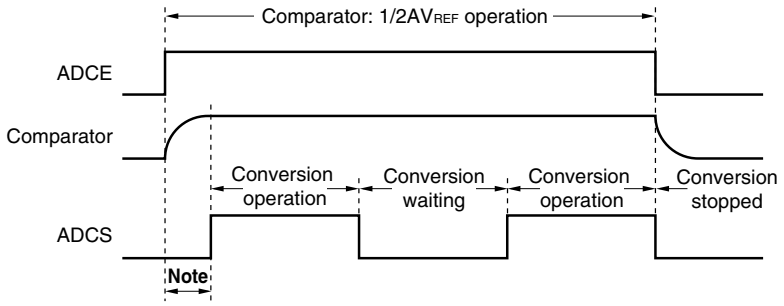
Notes 1. For details of FR2 to FR0, LV1, LV0, and A/D conversion, see **Table 10-2 A/D Conversion Time Selection**.

2. The operation of the comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

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 (comparator: 1/2AV$_{REF}$ operation, only comparator consumes power)
1	0	Setting prohibited
1	1	Conversion mode (comparator: 1/2AV$_{REF}$ operation)

Figure 10-5. Timing Chart When Comparator Is Used



Note To stabilize the internal circuit, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 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) 2.7 V ≤ AV_{REF} ≤ 5.5 V

A/D Converter Mode Register (ADM)					Conversion Time Selection				Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CLK} = 2 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	
0	0	0	0	0	264/f_{CLK}	Setting prohibited	26.4 μs	13.2 μs	f_{CLK}/12
0	0	1	0	0	176/f_{CLK}		17.6 μs	8.8 μs[Note]	f_{CLK}/8
0	1	0	0	0	132/f_{CLK}		13.2 μs	6.6 μs[Note]	f_{CLK}/6
0	1	1	0	0	88/f_{CLK}		8.8 μs[Note]	Setting prohibited	f_{CLK}/4
1	0	0	0	0	66/f_{CLK}	33.0 μs	6.6 μs[Note]		f_{CLK}/3
1	0	1	0	0	44/f_{CLK}	22.0 μs	Setting prohibited		f_{CLK}/2
1	1	1	0	0	22/f_{CLK}	11.0 μs[Note]			f_{CLK}
Other than above					Setting prohibited				

Note This can be set only when 4.0 V ≤ AV_{REF} ≤ 5.5 V.

Caution Set the conversion times with the following conditions.

- **4.0 V ≤ AV_{REF} ≤ 5.5 V: f_{AD} = 0.6 to 3.6 MHz**
- **2.7 V ≤ AV_{REF} < 4.0 V: f_{AD} = 0.6 to 1.8 MHz**

(2) 2.3 V ≤ AV_{REF} ≤ 5.5 V

A/D Converter Mode Register (ADM)					Conversion Time Selection			Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	
0	0	0	0	1	480/f_{CLK}	Setting prohibited	Setting prohibited	f_{CLK}/12
0	0	1	0	1	320/f_{CLK}		64.0 μs[Note 1]	f_{CLK}/8
0	1	0	0	1	240/f_{CLK}		48.0 μs[Note 1]	f_{CLK}/6
0	1	1	0	1	160/f_{CLK}		32.0 μs	f_{CLK}/4
1	0	0	0	1	120/f_{CLK}	60.0 μs	24.0 μs[Note 2]	f_{CLK}/3
1	0	1	0	1	80/f_{CLK}	40.0 μs	16.0 μs[Note 3]	f_{CLK}/2
1	1	1	0	1	40/f_{CLK}	20.0 μs[Note 3]	Setting prohibited	f_{CLK}
Other than above					Setting prohibited			

Notes 1. This can be set only when 2.3 V ≤ AV_{REF} < 2.7 V.

2. This can be set only when 2.7 V ≤ AV_{REF} ≤ 5.5 V.

3. This can be set only when 4.0 V ≤ AV_{REF} ≤ 5.5 V.

Cautions 1. Set the conversion times with the following conditions.

- **4.0 V ≤ AV_{REF} ≤ 5.5 V: f_{AD} = 1.2 to 3.6 MHz**
- **2.7 V ≤ AV_{REF} < 4.0 V: f_{AD} = 1.2 to 1.8 MHz**
- **2.3 V ≤ AV_{REF} < 2.7 V: f_{AD} = 0.6 to 1.44 MHz**

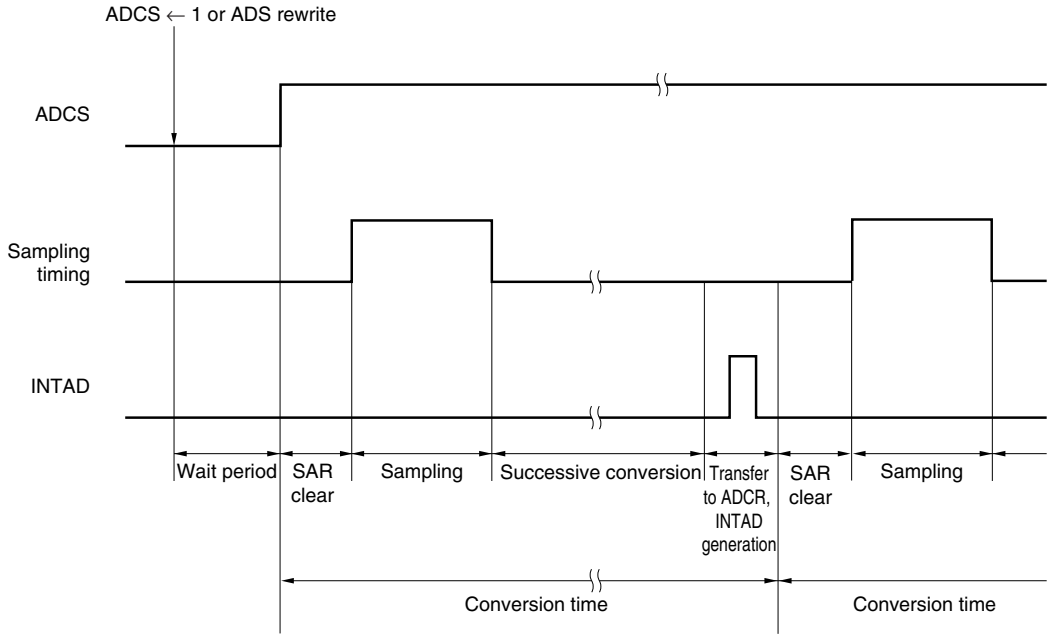
2. When rewriting FR2 to FR0, LV1, and LV0 to other than the same data, stop A/D conversion once (ADCS = 0) beforehand.

3. Change LV1 and LV0 from the default value, when 2.3 V ≤ AV_{REF} < 2.7 V.

4. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 10-6. A/D Converter Sampling and A/D Conversion Timing



(3) 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register. The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH.

ADCR can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 10-7. 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 the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.

(4) 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored.

ADCRH can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-8. Format of 8-Bit A/D Conversion Result Register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to the A/D converter mode register (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 10-9. Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	0	0	ADS2	ADS1	ADS0

ADISS	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	ANI0	P20/ANI0 pin
0	0	0	1	ANI1	P21/ANI1 pin
×	0	1	0	ANI2	P22/ANI2 pin
×	0	1	1	ANI3	P23/ANI3 pin
×	1	0	0	ANI4	P24/ANI4 pin
×	1	0	1	ANI5	P25/ANI5 pin
×	1	1	0	ANI6	P26/ANI6 pin
×	1	1	1	ANI7	P27/ANI7 pin

Cautions 1. Be sure to clear bits 3 to 6 to "0".

2 Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 (PM2).

3. Do not set the pin that is set by ADPC as digital I/O by ADS.

Remark ×: don't care

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

This register switches the ANI0/P20 to ANI7/P27 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 10-10. Format of A/D Port Configuration Register (ADPC)

Address: F0017H After reset: 10H R/W

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

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	Analog Input (A)/digital I/O (D) switching							
					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
0	0	0	0	1	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	D	D	D	D	D	D
0	0	1	1	1	A	D	D	D	D	D	D	D
0	1	0	0	0	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D
Other than above					Setting prohibited							

Cautions 1. Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 (PM2).

2. Do not set the pin that is set by ADPC as digital I/O by ADS.

3. When using all ANI0/P20 to ANI7/P27 pins as digital I/O (D), the setting can be done by ADPC4 to ADPC0 = either 01000 or 10000.

(7) Port mode registers 2 (PM2)

When using the ANI0/P20 to ANI7/P27 pins for analog input port, set PM20 to PM27 to 1. The output latches of P20 to P27 at this time may be 0 or 1.

If PM20 to PM27 are set to 0, they cannot be used as analog input port pins.

PM2 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. Format of Port Mode Registers 2 (PM2)

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

PM2n	P2n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

ANI0/P20 to ANI7/P27 pins are as shown below depending on the settings of ADPC, ADS, and PM2.

Table 10-3. Setting Functions of ANI0/P20 to ANI7/P27 Pins

ADPC	PM2	ADS	ANI0/P20 to ANI7/P27 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.	

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 bit 0 (ADCE) of the A/D converter mode register (ADM) to 1 to start the operation of the comparator.

<3> Set channels for A/D conversion to analog input by using bits the A/D port configuration register (ADPC) and set to input mode by using port mode registers 2 (PM2).

<4> Set A/D conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM.

<5> Select one channel for A/D conversion using the analog input channel specification register (ADS).

<6> Start the conversion operation by setting bit 7 (ADCS) of ADM to 1.
(<7> to <13> are operations performed by hardware.)

<7> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<8> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<9> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{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.

- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

<12> Comparison is continued in this way up to bit 0 of SAR.

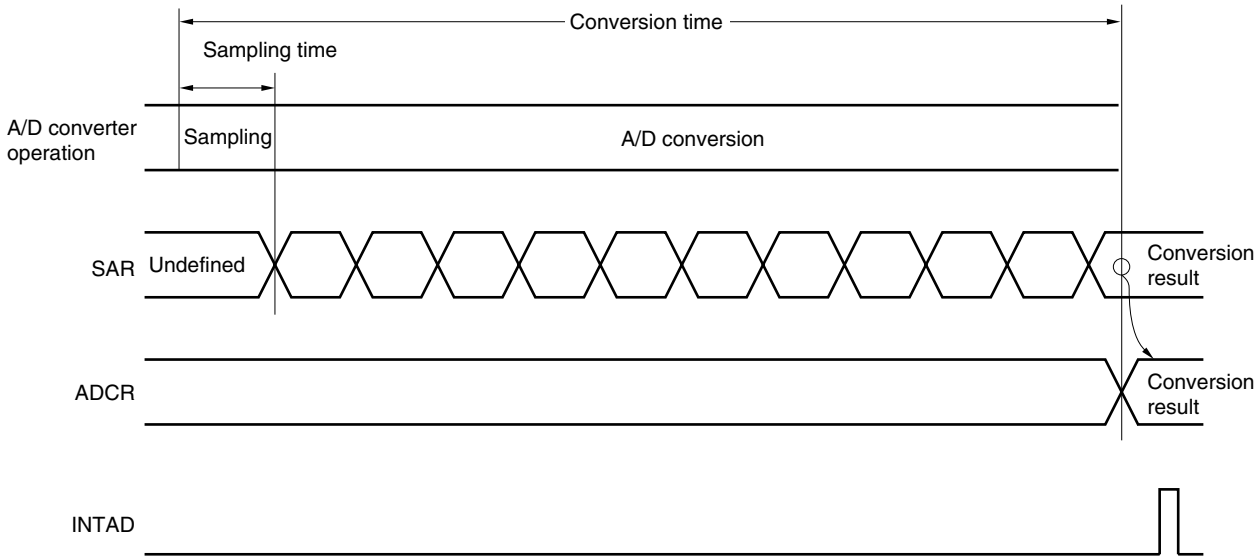
<13> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<14> Repeat steps <7> to <13>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCE = 1, start from <6>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start <6>. To change a channel of A/D conversion, start from <5>.

Caution Make sure the period of <2> to <6> is 1 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 the A/D converter mode register (ADM) is reset (0) by software.

If a write operation is performed to the analog input channel specification register (ADS) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (1), conversion starts again from the beginning.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

10.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7) 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

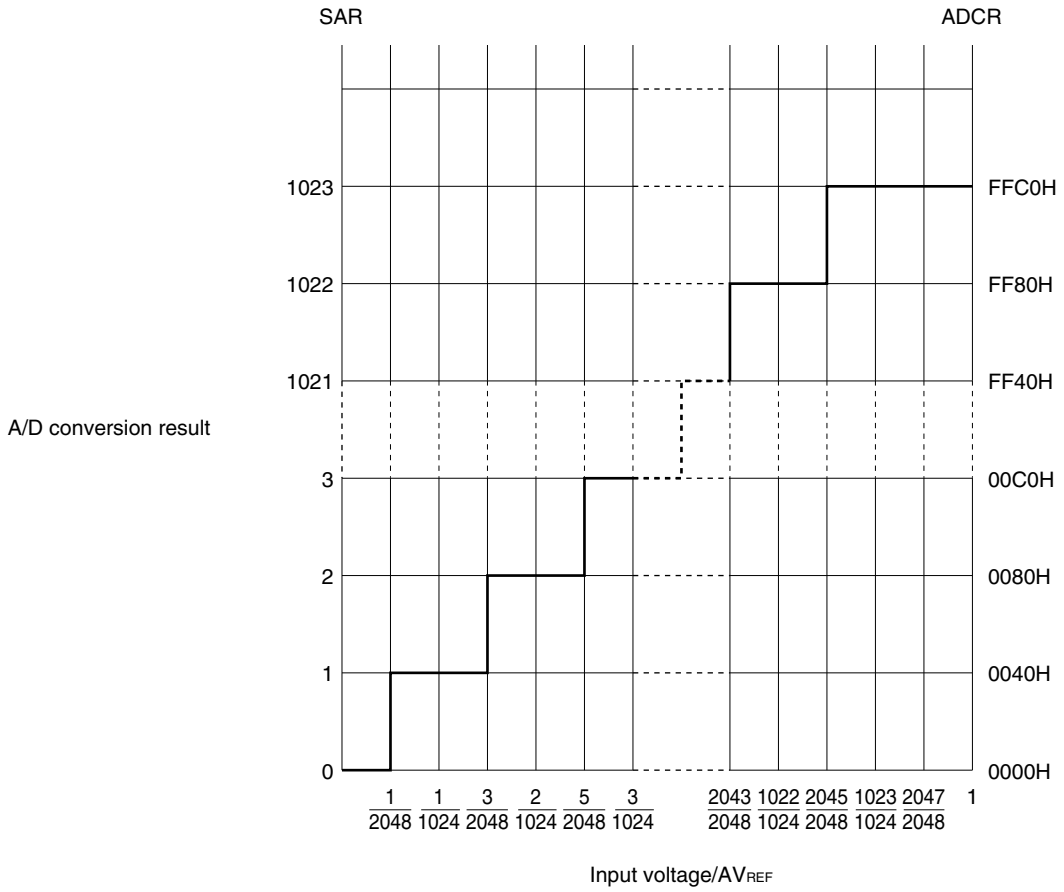
AV_{REF}: AV_{REF} pin voltage

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



10.4.3 A/D converter operation mode

The operation mode of the A/D converter is the select mode. One channel of analog input is selected from ANI0 to ANI7 by the analog input channel specification register (ADS) and A/D conversion is executed.

(1) A/D conversion operation

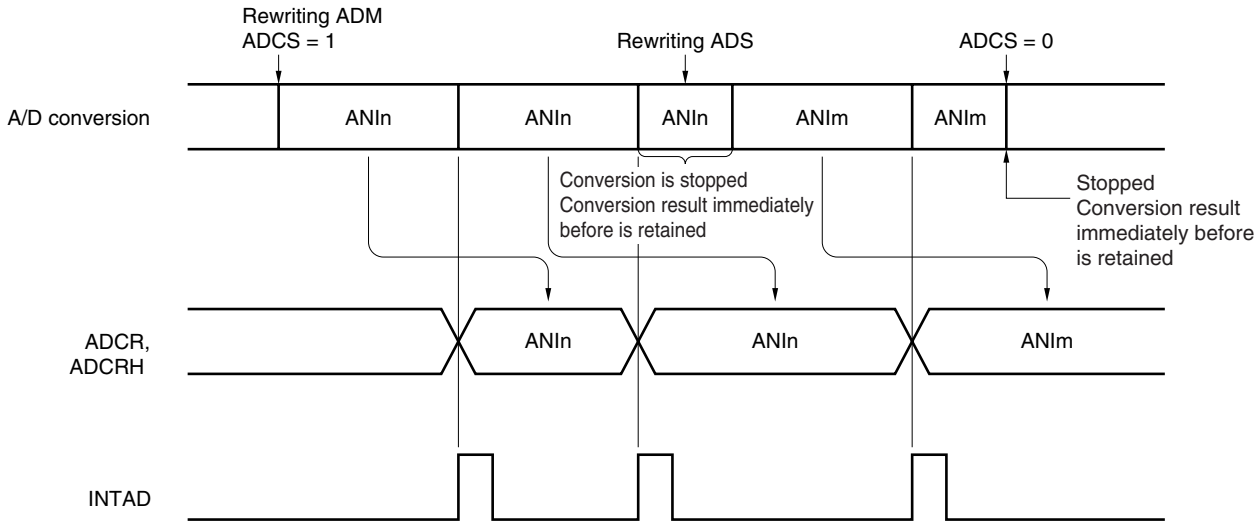
By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, the next A/D conversion operation is immediately started.

If ADS is rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning.

If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped. At this time, the conversion result immediately before is retained.

Figure 10-14. A/D Conversion Operation



Remarks 1. n = 0 to 7

2. m = 0 to 7

The setting methods are described below.

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1.
<2> Set bit 0 (ADCE) of the A/D converter mode register (ADM) to 1.
<3> Set the channel to be used in the analog input mode by using bits 4 to 0 (ADPC4 to ADPC0) of the A/D port configuration register (ADPC) and bits 7 to 0 (PM27 to PM20) of port mode register 2 (PM2).
<4> Select conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of ADM.
<5> Select a channel to be used by using bits 7 and 2 to 0 (ADISS, ADS2 to ADS0) of the analog input channel specification register (ADS).
<6> Set bit 7 (ADCS) of ADM to 1 to start A/D conversion.
<7> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<8> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Change the channel>

<9> Change the channel using bits 7 and 2 to 0 (ADISS, ADS2 to ADS0) of ADS to start A/D conversion.
<10> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<11> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Complete A/D conversion>

<12> Clear ADCS to 0.
<13> Clear ADCE to 0.
<14> Clear bit 5 (ADCEN) of peripheral enable register 0 (PER0)

Cautions 1. Make sure the period of <2> to <6> is 1 μs or more.
2. <2> may be done between <3> and <5>.
3. <2> can be omitted. However, ignore data of the first conversion after <6> 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.

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 10-15. Overall Error

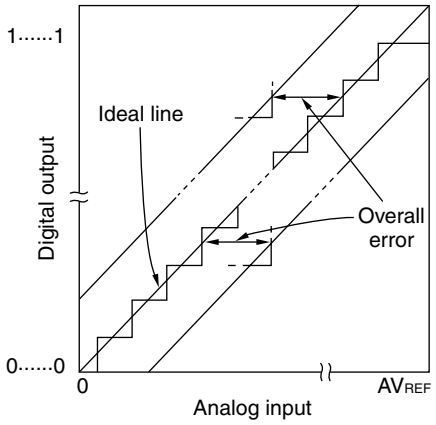
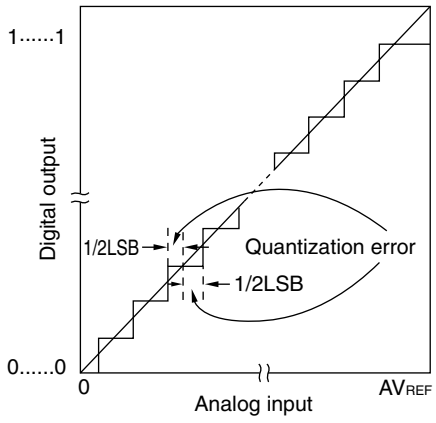


Figure 10-16. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 10-17. Zero-Scale Error

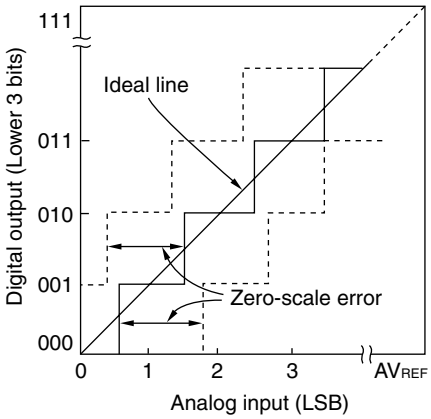


Figure 10-18. Full-Scale Error

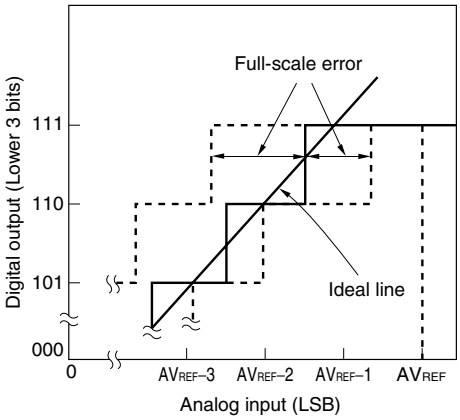


Figure 10-19. Integral Linearity Error

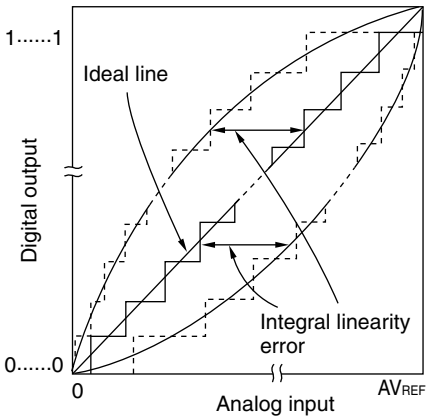
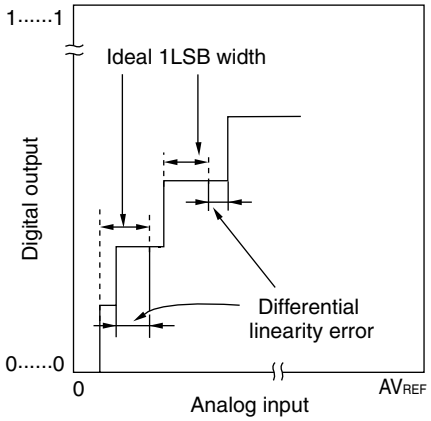


Figure 10-20. Differential Linearity Error



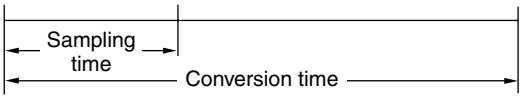
(8) Conversion time

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

The A/D converter stops operating in the STOP mode. At this time, the operating current can be reduced by clearing bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) to 0.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1L (IF1L) to 0 and start operation.

(2) Input range of ANI0 to ANI7

Observe the rated range of the ANI0 to ANI7 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.

(3) 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.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REF} pin and pins ANI0 to ANI7.

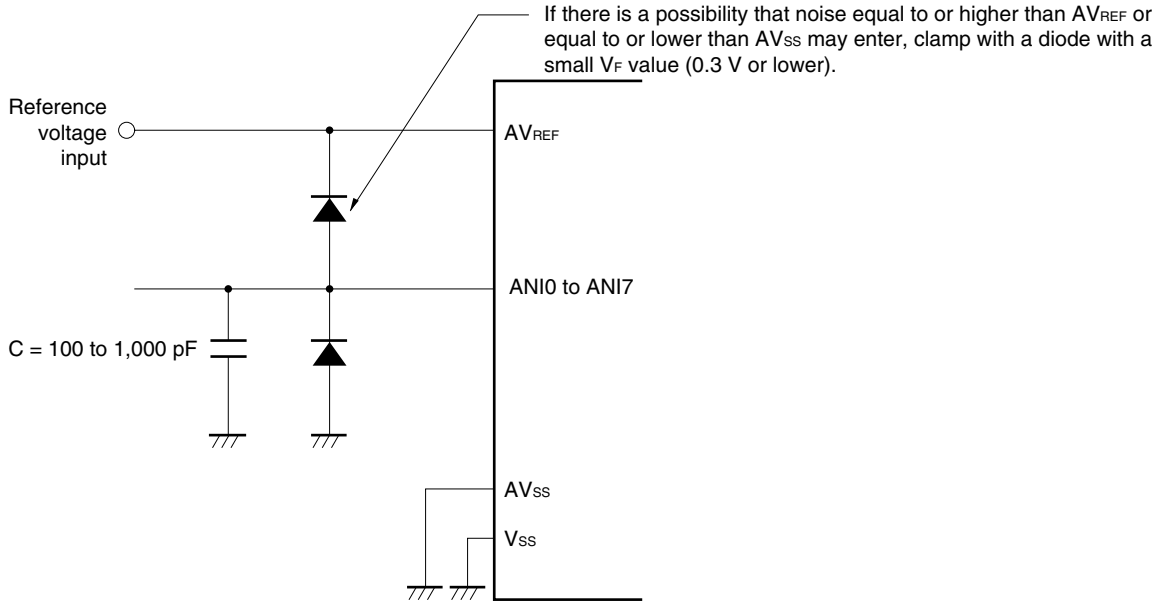
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<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 10-21 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 10-21. Analog Input Pin Connection



(5) ANI0/P20 to ANI7/P27

<1> The analog input pins (ANI0 to ANI7) are also used as input port pins (P20 to P27).
When A/D conversion is performed with any of ANI0 to ANI7 selected, do not access P20 to P27 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to select pins used as P20 to P27 starting with the ANI0/P20 that is the furthest from AV_{REF}.

<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.

(6) Input impedance of ANI0 to ANI7 pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 10 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI7 pins (see **Figure 10-21**).

(7) AV_{REF} pin input impedance

A series resistor string of several tens of kΩ is connected between the AV_{REF} and AV_{SS} pins.
Therefore, if the output impedance of the reference voltage source is high, this will result in a series connection to the series resistor string between the AV_{REF} and AV_{SS} pins, resulting in a large reference voltage error.

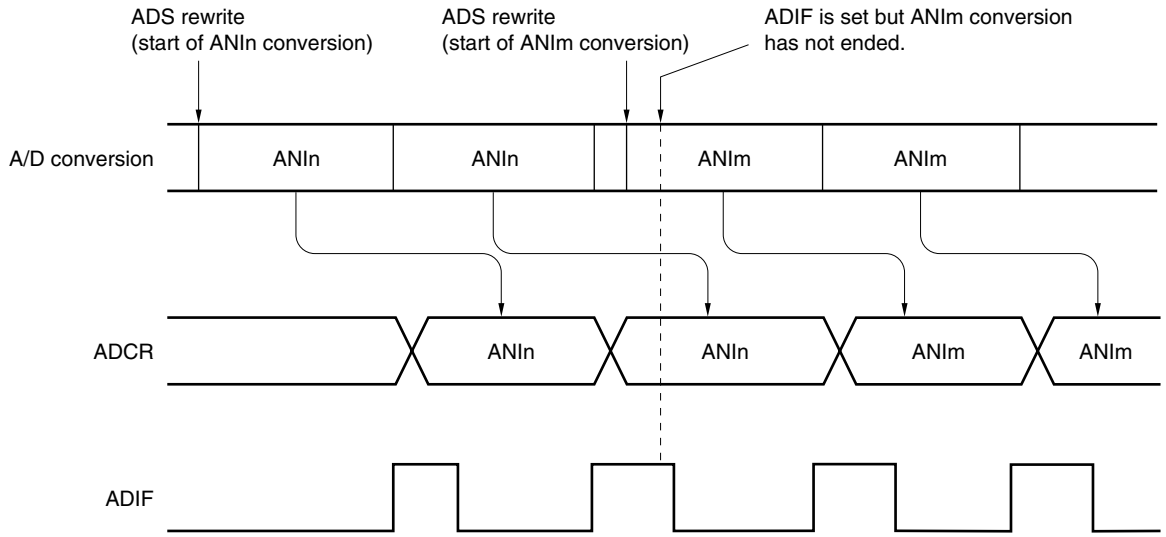
(8) Interrupt request flag (ADIF)

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 10-22. Timing of A/D Conversion End Interrupt Request Generation



Remarks 1. n = 0 to 7

2. m = 0 to 7

(9) 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, or if the ADCS bit is set to 1 with the ADCE bit = 0. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(10) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to the A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR and ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using a timing other than the above may cause an incorrect conversion result to be read.

(11) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 10-23. Internal Equivalent Circuit of ANIn Pin

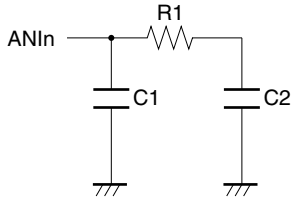


Table 10-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	R1	C1	C2
4.0 V ≤ V_{DD} ≤ 5.5 V	8.1 kΩ	8 pF	5 pF
2.7 V ≤ V_{DD} < 4.0 V	31 kΩ	8 pF	5 pF
2.3 V ≤ V_{DD} < 2.7 V	381 kΩ	8 pF	5 pF

Remarks 1. The resistance and capacitance values shown in Table 10-4 are not guaranteed values.

2. n = 0 to 7

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/KE3 is as shown below (channels 2 and 3 of unit 1 are dedicated to UART3 (supporting LIN-bus)).

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	–
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	–	–
	1	–	–	–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

(Example of combination) When “UART1” is used for channels 2 and 3 of unit 0, CSI10 and IIC10 cannot be used, but CSI00 or UART0 can be used.

11.1 Functions of Serial Array Unit

Each serial interface supported by the 78K0R/KE3 has the following features.

11.1.1 3-wire serial I/O (CSI00, CSI10)

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- Level setting of transmit/receive data

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

11.1.2 UART (UART0, UART1, UART3)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. It transmits or receives data in asynchronization with the party of communication (by using an internal baud rate). Full-duplex UART communication can be realized by using two channels, one dedicated to transmission (even channel) and the other to reception (odd channel).

[Data transmission/reception]

- Data length of 5, 7, or 8 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

The LIN-bus is accepted in UART3 (2 and 3 channels of unit 1)

[LIN-bus functions]

- Wakeup signal detection
- Sync break field (SBF) detection
- Sync field measurement, baud rate calculation

} External interrupt (INTP0) or timer array unit (TAU) is used.

11.1.3 Simplified I^2C (IIC10)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I^2C 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 and ACK detection functions
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Parity error (ACK error)

* [Functions not supported by simplified I^2C]
  - Slave transmission, slave reception
  - Arbitration loss detection function
  - Wait detection functions

Remark To use an I^2C bus of full function, see **CHAPTER 12 SERIAL INTERFACE IIC0**.

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, SCK10 pins (for 3-wire serial I/O), SCL10 pin (for simplified I^2C)
Serial data input	SI00, SI10 pins (for 3-wire serial I/O), RxD0, RxD1 pins (for UART), RxD3 pin (for UART supporting LIN-bus)
Serial data output	SO00, SO10 pins (for 3-wire serial I/O), TxD0, TxD1 pins (for UART), TxD3 pin (for UART supporting LIN-bus), output controller
Serial data I/O	SDA10 pin (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) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn) • Port input mode registers 0 (PIM0) • Port output mode registers 0 (POM0) • Port mode registers 0, 1 (PM0, PM1) • Port registers 0, 1 (P0, P1)

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)
- IIC10 communication … SIO10 (IIC10 data register)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 12, 13
p: CSI number (p = 00, 10), q: UART number (q = 0, 1, 3)

Figure 11-1 shows the block diagram of serial array unit 0.

Figure 11-1. Block Diagram of Serial Array Unit 0

<R>

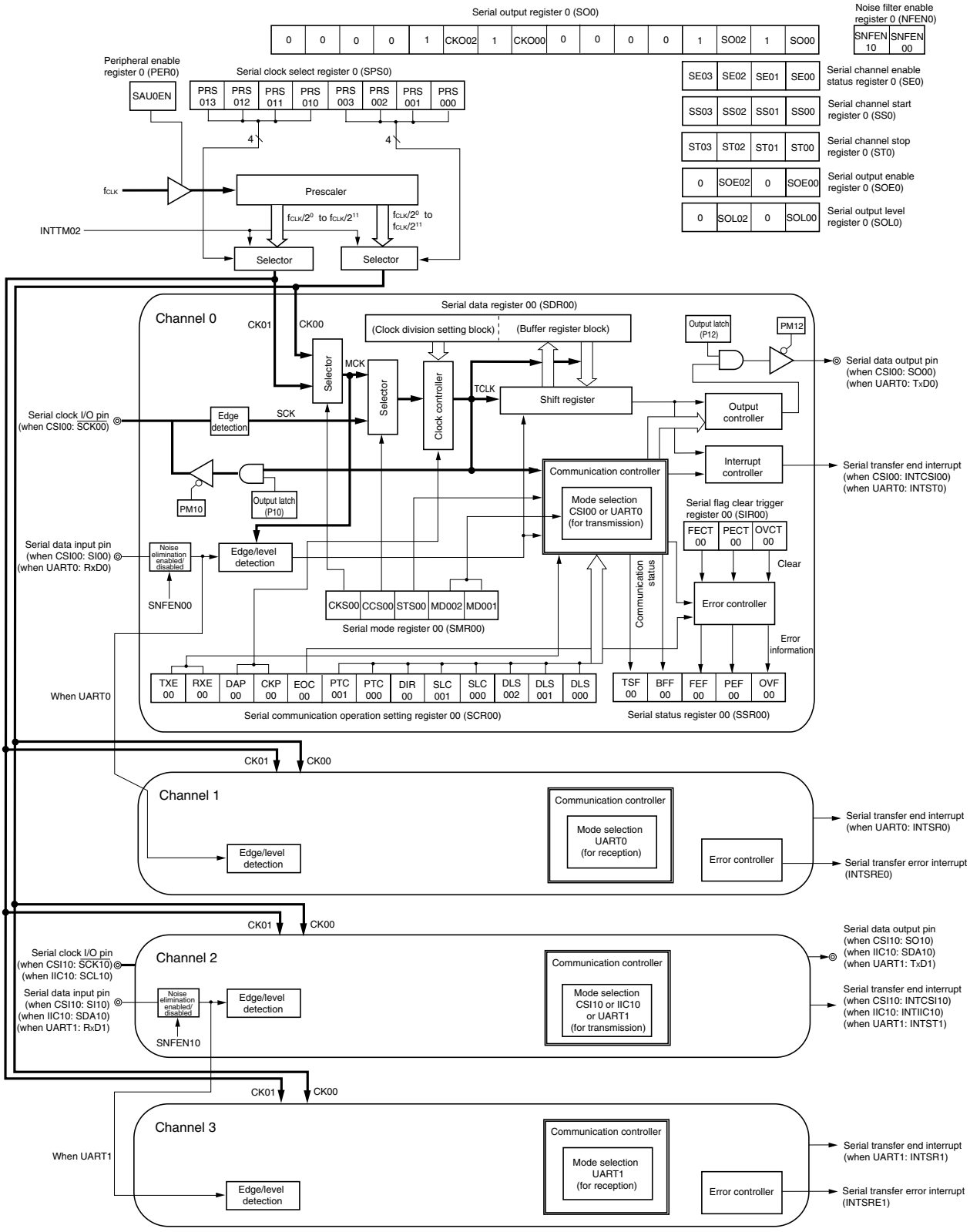
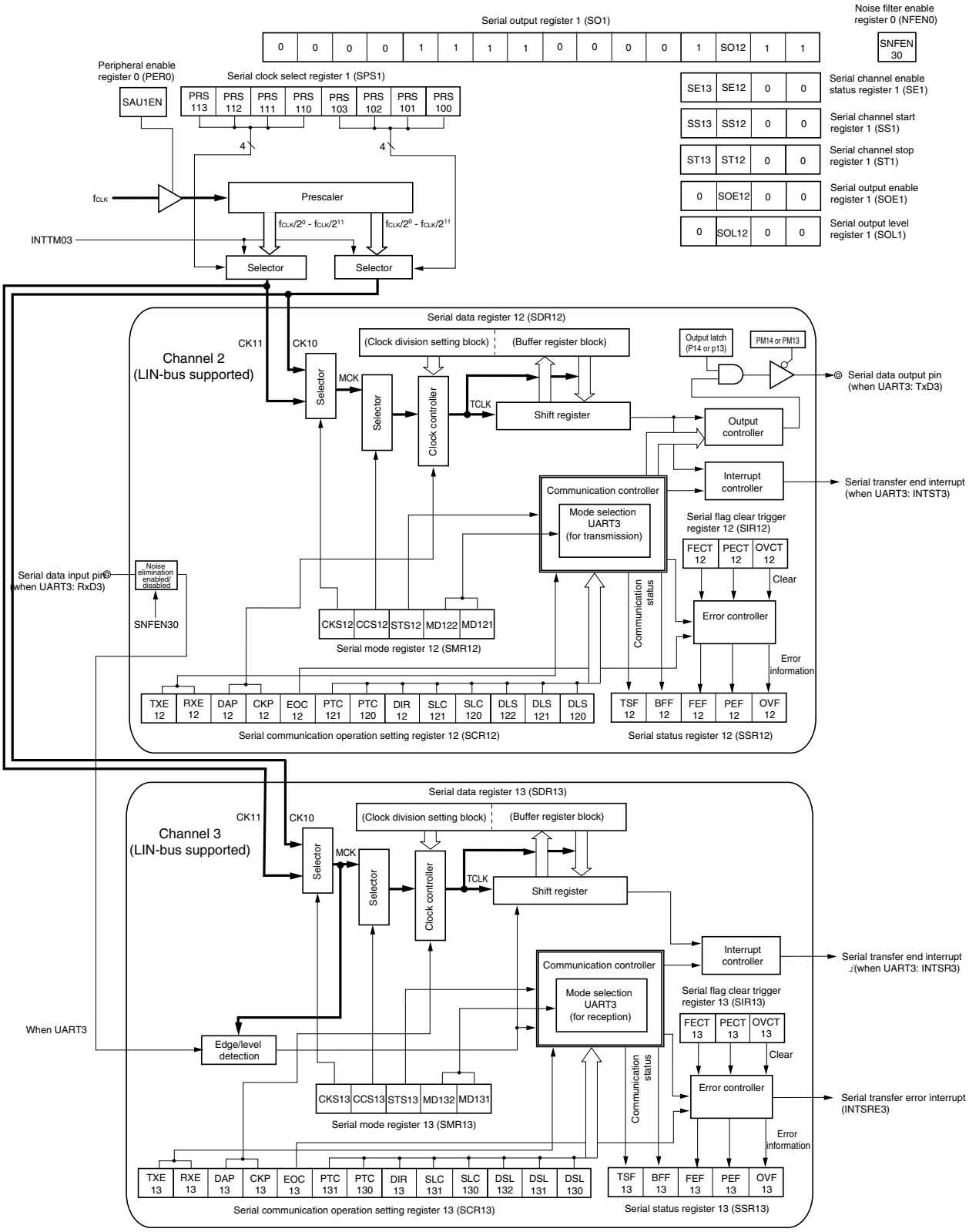


Figure 11-2 shows the block diagram of serial array unit 1.

Figure 11-2. Block Diagram of Serial Array Unit 1

<R>



(1) Shift register

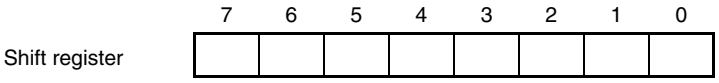
This is an 8-bit register that converts parallel data into serial data or vice versa.

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 8 bits of serial data register mn (SDRmn).



(2) Lower 8 bits of the serial data register mn (SDRmn)

SDRmn is the transmit/receive data register (16 bits) of channel n. Bits 7 to 0 function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (MCK).

When data is received, parallel data converted by the shift register is stored in the lower 8 bits. When data is to be transmitted, set transmit to be transferred to the shift register to the lower 8 bits.

The data stored in the lower 8 bits of this register is as follows, depending on the setting of bits 0 to 2 (DLSmn0 to DLSmn2) of the SCRmn register, regardless of the output sequence of the data.

- 5-bit data length (stored in bits 0 to 4 of SDRmn register) (settable in UART mode only)
- 7-bit data length (stored in bits 0 to 6 of SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of SDRmn register)

SDRmn can be read or written in 16-bit units.

The lower 8 bits of SDRmn of SDRmn can be read or written[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)
- IIC10 communication … SIO10 (IIC10 data register)

Reset signal generation clears this register to 0000H.

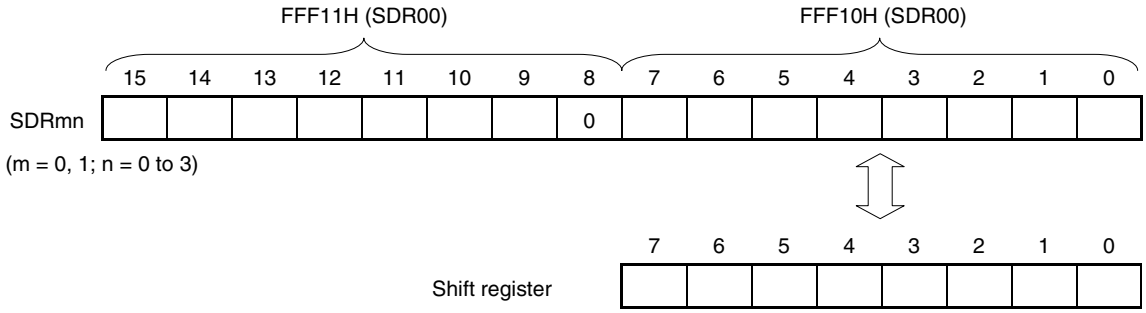
Note Writing in 8-bit units is prohibited when the operation is stopped (SEmn = 0).

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, 12, 13
p: CSI number (p = 00, 10), q: UART number (q = 0, 1, 3)

Figure 11-3. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03), FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13) 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, 12, 13
p: CSI number (p = 00, 10), q: UART number (q = 0, 1, 3)

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)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 0 (PIM0)
- Port output mode registers 0 (POM0)
- Port mode registers 0, 1 (PM0, PM1)
- Port registers 0, 1 (P0, P1)

Remark m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 3)
mn = 00 to 03, 12, 13

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When 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	IIC0EN	SAU1EN	SAU0EN	0	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 input switch control register (ISC), noise filter enable register (NFEN0), port input mode register (PIM0), port output mode register (POM0), port mode registers (PM0, PM1), and port registers (P0, P1)).

2. After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

3. Be sure to clear bits 1 and 6 of 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, INTTM03 if m = 1 Note 2				
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 and INTTM03 for the operation clock, also stop the timer array unit (TAU) (TT0 = 00FFH).

2. SAU can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by setting the TIS02 (if m = 0) and TIS03 (if m = 1) bits of the TIS0 register of TAU to 1, selecting $f_{SUB}/4$ for the input clock, and selecting INTTM02 and INTTM03 using the SPSm register. When changing f_{CLK}, however, SAU and TAU must be stopped as described in Note 1 above.

Cautions 1. Be sure to clear bits 15 to 8 to "0".

2. After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. f_{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 (MCK), specify whether the serial clock (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), F0154H, F0155H (SMR12), F0156H, F0157H (SMR13) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn	0	SISmn0	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operation clock (MCK) of channel n
0	Prescaler output clock CKm0 set by PRS register
1	Prescaler output clock CKm1 set by PRS register
Operation clock MCK is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (TCLK) is generated.	

CCSmn	Selection of transfer clock (TCLK) of channel n
0	Divided operation clock MCK specified by CKSmn bit
1	Clock input from SCK pin (slave transfer in CSI mode)
Transfer clock TCLK is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of MCK is set by the higher 7 bits of the SDRmn register.	

STSmn	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, 12, 13

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
F0154H, F0155H (SMR12), F0156H, F0157H (SMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn	0	SIS mn0	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MD mn2	MD mn1	Setting of operation mode of channel n
0	0	CSI mode
0	1	UART mode
1	0	Simplified I^2C mode
1	1	Setting prohibited

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt
For successive transmission, the next transmit data is written by setting MDmn0 to 1 when SDRmn data has run out.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 12, 13

(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.

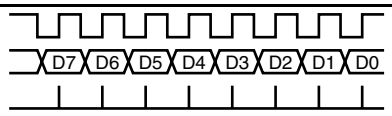
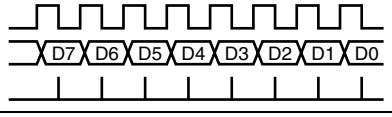
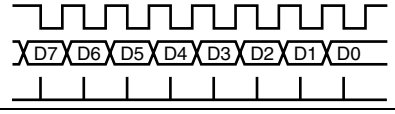
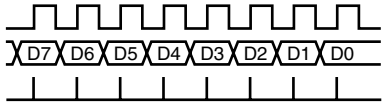
Figure 11-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F015CH, F015DH (SCR12), F015EH, F015FH (SCR13)

After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	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
0	0	
0	1	
1	0	
1	1	
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, 12, 13, p: CSI number (p = 00, 10)

Figure 11-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F015CH, F015DH (SCR12), F015EH, F015FH (SCR13) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

EOC mn	Selection of masking of error interrupt signal (INTSREx (x = 0, 1, 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. 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.	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.		

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, 12, 13

Figure 11-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (3/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F015CH, F015DH (SCR12), F015EH, F015FH (SCR13)

After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	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, 12, 13

(5) Higher 7 bits of the serial data register mn (SDRmn)

SDRmn is the transmit/receive data register (16 bits) of channel n. Bits 7 to 0 function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (MCK).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by the higher 7 bits of SDRmn is used as the transfer clock.

For the function of the lower 8 bits of SDRmn, see **11.2 Configuration of Serial Array Unit**.

SDRmn can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8 bits of SDRmn. When SDRmn is read during operation, 0 is always read.

Reset signal generation clears this register to 0000H.

Figure 11-8. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03), FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13)

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]							Setting of division ratio of operation clock (MCK)
0	0	0	0	0	0	0	MCK/2
0	0	0	0	0	0	1	MCK/4
0	0	0	0	0	1	0	MCK/6
0	0	0	0	0	1	1	MCK/8
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	MCK/254
1	1	1	1	1	1	1	MCK/256

Cautions 1. Be sure to clear bit 8 to "0".

2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

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, 12, 13

(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), F0144H, F0145H (SSR12), F0146H, F0147H (SSR13)

After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn	PEF mn	OVF mn

TSF mn	Communication status indication flag of channel n
0	Communication is not under execution.
1	Communication is under execution.
Because this flag is an updating flag, it is automatically cleared when the communication operation is completed. This flag is cleared also when the STmn/SSmn bit is set to 1.	

BFF mn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
This is an updating flag. It is automatically cleared when transfer from the SDRmn register to the shift register is completed. During reception, it is automatically cleared when data has been read from the SDRmn register. This flag is cleared also when the STmn/SSmn bit is set to 1. This flag is automatically set if transmit data is written to the SDRmn register when the TXEmn bit of the SCRmn register = 1 (transmission or reception mode in each communication mode). It is automatically set if receive data is stored in the SDRmn register when the RXEmn bit of the SCRmn register = 1 (transmission or reception mode in each communication mode). It is also set in case of a reception error. If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVFmn = 1) is detected.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 12, 13

Figure 11-9. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), F0144H, F0145H (SSR12), F0146H, F0147H (SSR13)

After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn	PEF mn	OVF mn

FEF mn	Framing error detection flag of channel n
0	No error occurs.
1	A framing error occurs during UART reception. <Framing error cause> A framing error occurs if the stop bit is not detected upon completion of UART reception.
This is a cumulative flag and is not cleared until 1 is written to the FECTmn bit of the SIRmn register.	

PEF mn	Parity error detection flag of channel n
0	Error does not occur.
1	A parity error occurs during UART reception or ACK is not detected during I^2C transmission. <Parity error cause> • A parity error occurs if the parity of transmit data does not match the parity bit on completion of UART reception. • ACK is not detected if the ACK signal is not returned from the slave in the timing of ACK reception during I^2C transmission.
This is a cumulative flag and is not cleared until 1 is written to the PECTmn bit of the SIRmn register.	

OVF mn	Overrun error detection flag of channel n
0	No error occurs.
1	An overrun error occurs. <Causes of overrun error> • Receive data stored in the SDRmn register is not read and transmit data is written or the next receive data is written. • Transmit data is not ready for slave transmission or reception in the CSI mode.
This is a cumulative flag and is not cleared until 1 is written to the OVCTmn bit of the SIRmn register.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 12, 13

(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), F014CH, F014DH (SIR12), F014EH, F014FH (SIR13) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FEC Tmn	PEC Tmn	OVC Tmn

FEC Tmn	Clear trigger of framing error of channel n
0	No trigger operation
1	Clears the FEFmn bit of the SSRmn register to 0.

PEC Tmn	Clear trigger of parity error flag of channel n
0	No trigger operation
1	Clears the PEFmn bit of the SSRmn register to 0.

OVC Tmn	Clear trigger of overrun error flag of channel n
0	No trigger operation
1	Clears the OVFmn bit of the SSRmn register to 0.

Caution Be sure to clear bits 15 to 3 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 12, 13

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 0 (SSm), the corresponding bit of this register is set to 1.

When 1 is written a bit of serial channel stop register 0 (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 After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE03	SE02	SE01	SE00

Address: F0160H, F0161H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE1	0	0	0	0	0	0	0	0	0	0	0	0	SE13	SE12	0	0

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, 12, 13

(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 After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

Address: F0162H, F0163H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13	SS12	0	0

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets SEmn to 1 and enters the communication wait status (if a communication operation is already under execution, the operation is stopped and the start condition is awaited).

Caution Be sure to clear bits 15 to 4 of SS0, and bits 15 to 4, 1 and 0 of SS1 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 12, 13

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 After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00

Address: F0164H, F0165H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST1	0	0	0	0	0	0	0	0	0	0	0	0	ST13	ST12	0	0

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 of ST0, and bits 15 to 4, 1 and 0 of ST1 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 12, 13

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	0	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	SOE12	0	0

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 and 1 of SOE0, and bits 15 to 3, 1 and 0 of SOE1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

(12) Serial output register m (SOm)

SOm is a buffer register for serial output of each channel.

The value of bit n of this register is output from the serial data output pin of channel n.

The value of bit (n + 8) of this register is output from the serial clock output pin of channel n.

SOmn of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

CKOmn of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of CKOmn can be changed only by a serial communication operation.

<R> To use the P02/SO10/TxD1, P03/SI10/SDA10/RxD1, P04/$\overline{\text{SCK10}}$/SCL10, P10/$\overline{\text{SCK00}}$, P12/SO00/TxD0, or P13/TxD3 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	CKO 02	1	CKO 00	0	0	0	0	1	SO 02	1	SO 00

Address: F0168H, F0169H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	1	0	0	0	0	1	SO 12	1	1

CKO mn	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

SO mn	Serial data output of channel n
0	Serial data output value is “0”.
1	Serial data output value is “1”.

Caution Be sure to set bits 11, 9, 3 and 1 of SO0, and bits 11 to 8, 3, 1 and 0 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, 2), mn = 00, 02, 12

(13) Serial output level register m (SOLm)

SOLm is a register that is used to set inversion of the data output level of each channel.

This register can be set only in the UART mode. Be sure to set 0000H in the CSI mode and simplifies I^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 After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL02	0	SOL00

Address: F0174H, F0175H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL12	0	0

SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution Be sure to clear bits 15 to 3 and 1 of SOL0, and bits 15 to 3, 1 and 0 of SOL1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

(14) Input switch control register (ISC)

ISC is used to realize a LIN-bus communication operation by UART3 in coordination with an external interrupt and the timer array unit.

When bit 0 is set to 1, the input signal of the serial data input (RxD3) pin is selected as an external interrupt (INTP0) that can be used to detect a wakeup signal.

When bit 1 is set to 1, the input signal of the serial data input (RxD3) pin is selected as a timer input, so that the pulse widths of a sync break field and a sync field can be measured by the timer.

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

Reset signal generation clears this register to 00H.

Figure 11-17. Format of Input Switch Control Register (ISC)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit
0	Not uses the input signal (normal operation).
1	Input signal of RxD3 pin is used as timer input (wakeup signal detection).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD3 pin as an external interrupt (to measure the pulse widths of the sync break field and sync field).

Caution Be sure to clear bits 7 to 2 to "0".

Remark Since the 78K0R/KE3 does not have the timer input pin on channel 7, normally the timer input on channel 7 cannot be used. When the LIN-bus communication function is used, select the input signal of the RxD3 pin by setting ISC1 to 1.

(15) Noise filter enable register 0 (NFEN0)

NFEN0 is used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for CSI or simplified I^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 11-18. Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0060H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	SNFEN30	0	0	0	SNFEN10	0	SNFEN00

SNFEN30	Use of noise filter of RxD3/P14 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN30 to 1 to use the RxD3 pin. Clear SNFEN30 to 0 to use the P14 pin.	

SNFEN10	Use of noise filter of RxD1/SDA10/SI10/P03 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN10 to 1 to use the RxD1 pin. Clear SNFEN10 to 0 to use the SDA10, SI10, and P03 pins.	

SNFEN00	Use of noise filter of RxD0/SI00/P11 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN00 to 1 to use the RxD0 pin. Clear SNFEN00 to 0 to use the SI00 and P11 pins.	

Caution Be sure to clear bits 7, 5 to 3, and 1 to "0".

(16) Port input mode registers 0 (PIM0)

This register set the input buffer of ports 0 in 1-bit units.

PIM0 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 Input Mode Registers 0 (PIM0)

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

PIM0n	P0n pin input buffer selection (n = 3, 4)
0	Normal input buffer
1	TTL input buffer

(17) Port output mode registers 0 (POM0)

This register set the output mode of ports 0 in 1-bit units.

POM0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 11-20. Format of Port Output Mode Registers 0 (POM0)

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

POM0n	P0n pin output buffer selection (n = 2 to 4)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

(18) Port mode registers 0, 1 (PM0, PM1)

These registers set input/output of ports 0 and 1 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, P13/TxD3 pins for serial data output or serial clock output, clear the PM02, PM03, PM04, PM10, PM12, and PM13 bits to 0, and set the output latches of P02, P03, P04, P10, P12, and P13 to 1.

When using the P03/SI10/RxD1/SDA10, P04/$\overline{\text{SCK10}}$/SCL10, P10/$\overline{\text{SCK00}}$, P11/SI00/RxD0, and P14/RxD3 pins for serial data input or serial clock input, set the PM03, PM04, PM10, PM11, and PM14 bits to 1. At this time, the output latches of P03, P04, P10, P11, and P14 may be 0 or 1.

PM0 and PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 11-21. Format of Port Mode Registers 0 and 1 (PM0, PM1)

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

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

<R> 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, P13/TxD3, or P14/RxD3 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 11-22. 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	IIC0EN	SAU1EN	SAU0EN		TAU0EN
	×	0	×	×	0/1	0/1	0	×

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 input switch control register (ISC), noise filter enable register (NFEN0), port input mode register (PIM0), port output mode register (POM0), port mode registers (PM0, PM1), and port registers (P0, P1)).

2. Be sure to clear bits 1 and 6 of 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 11-23. Each Register Setting When Stopping the Operation by Channels (1/2)

(a) Serial Channel Enable Status Register m (SEm) … This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

SE0

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	SE03 0/1	SE02 0/1	SE01 0/1	SE00 0/1

0: Operation stops

* The SE0 register is a read-only status register, whose operation is stopped by using the ST0 register.
With a channel whose operation is stopped, the value of CKO0n of the SO0 register can be set by software.

SE1

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	SE13 0/1	SE12 0/1	0	0

0: Operation stops

* The SE1 register is a read-only status register, whose operation is stopped by using the ST1 register.

(b) Serial channel stop register m (STm) … This register is a trigger register that is used to enable stopping communication/count by each channel.

ST0

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	ST03 0/1	ST02 0/1	ST01 0/1	ST00 0/1

1: Clears SE0n to 0 and stops the communication operation

* Because ST0n is a trigger bit, it is cleared immediately when SE0n = 0.

ST1

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	ST13 0/1	ST12 0/1	0	0

1: Clears SE1n to 0 and stops the communication operation

* Because ST1n is a trigger bit, it is cleared immediately when SE1n = 0.

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 11-23. Each Register Setting When Stopping the Operation by Channels (2/2)

(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	0	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	SOE12 0/1	0	0

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SO12 value of the SO1 register can be set by software.

(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	1	CKO00 0/1	0	0	0	0	1	SO02 0/1	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	1	0	0	0	0	1	SO12 0/1	1	1

1: Serial data output value is “1”

* When using pins corresponding to each channel as port function pins, set the corresponding SO12 bit to “1”.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

□ : 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, CSI10) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- Level setting of transmit/receive data

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

The channels supporting 3-wire serial I/O (CSI00, CSI10) are channels 0, 2 of SAU0.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	–
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	–	–
	1	–	–	–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

3-wire serial I/O (CSI00, CIS10) 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 that the 78K0R/KE3 outputs a transfer clock and transmits data to another device.

3-Wire Serial I/O	CSI00	CSI10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCK00, SO00	SCK10, SO10
Interrupt	INTCSI00	INTCSI10
	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. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz] Note f_{CLK}: System clock frequency	
Data phase	Selectable by DAP0n bit • DAP0n = 0: Data output starts from the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by CKP0n bit • CKP0n = 0: Forward • CKP0n = 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 27 ELECTRICAL SPECIFICATIONS**).

(1) Register setting

Figure 11-24. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI10)

(a) Serial output register 0 (SO0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 0/1	1	CKO00 0/1	0	0	0	0	1	SO02 0/1	1	SO00 0/1

Communication starts when these bits are 1 if the data phase is forward (CKP0n = 0). If the phase is reversed (CKP0n = 1), communication starts when these bits are 0.

(b) Serial output enable register 0 (SOE0) … 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
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	0	SOE00 0/1

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 0/1

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 0	0	0	0	0	0	STS0n 0	0	SIS00 0	1	0	0	MD0n2 0	MD0n1 0	MD0n0 0/1

Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n 1	RXE0n 0	DAP0n 0/1	CKP0n 0/1	0	EOC0n 0	PTC0n1 0	PTC0n0 0	DIR0n 0/1	0	SLC0n1 0	SLC0n0 0	0	DLS0n2 1	DLS0n1 1	DLS0n0 0/1

(f) Serial data register 0n (SDR0n) (lower 8 bits: SIOp)

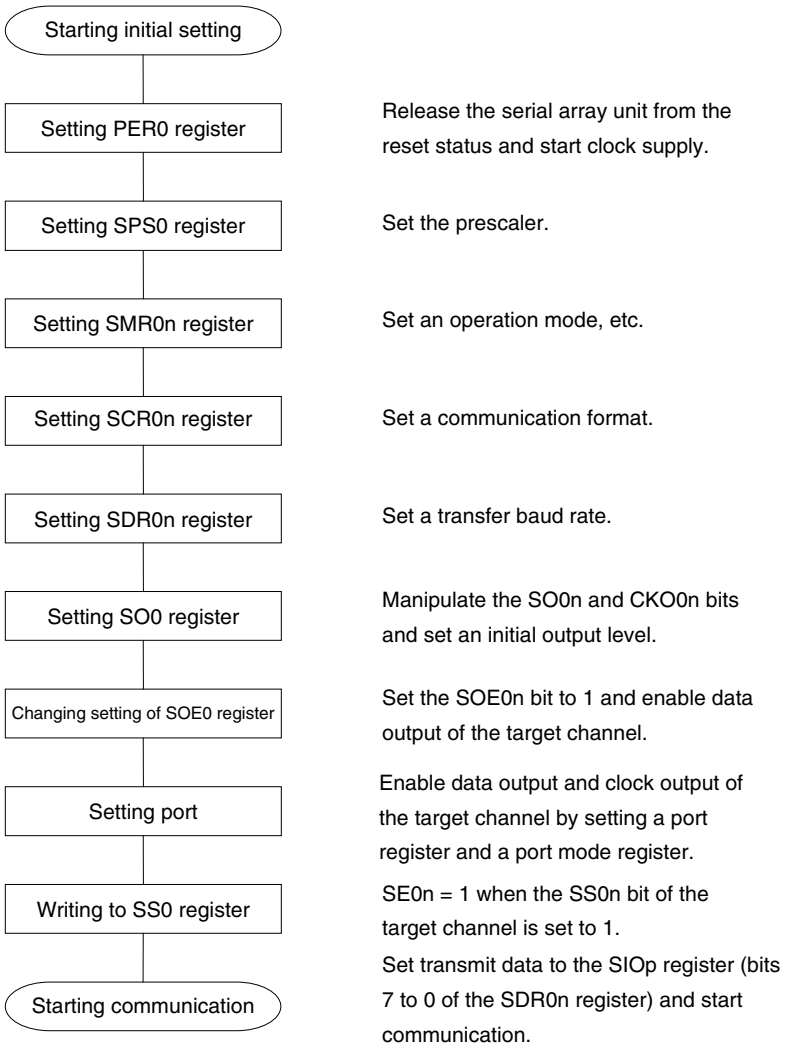
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	Baud rate setting							0	Transmit data setting							

SIOp

Remark n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)
□: 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

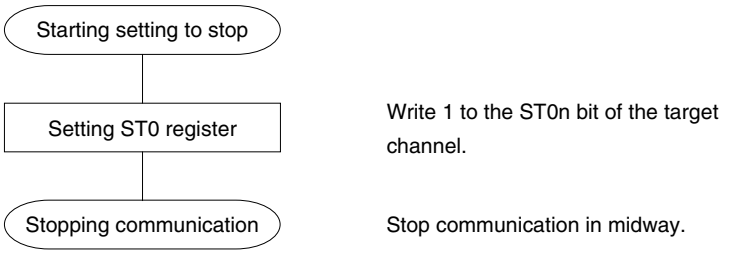
(2) Operation procedure

Figure 11-25. Initial Setting Procedure for Master Transmission



Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

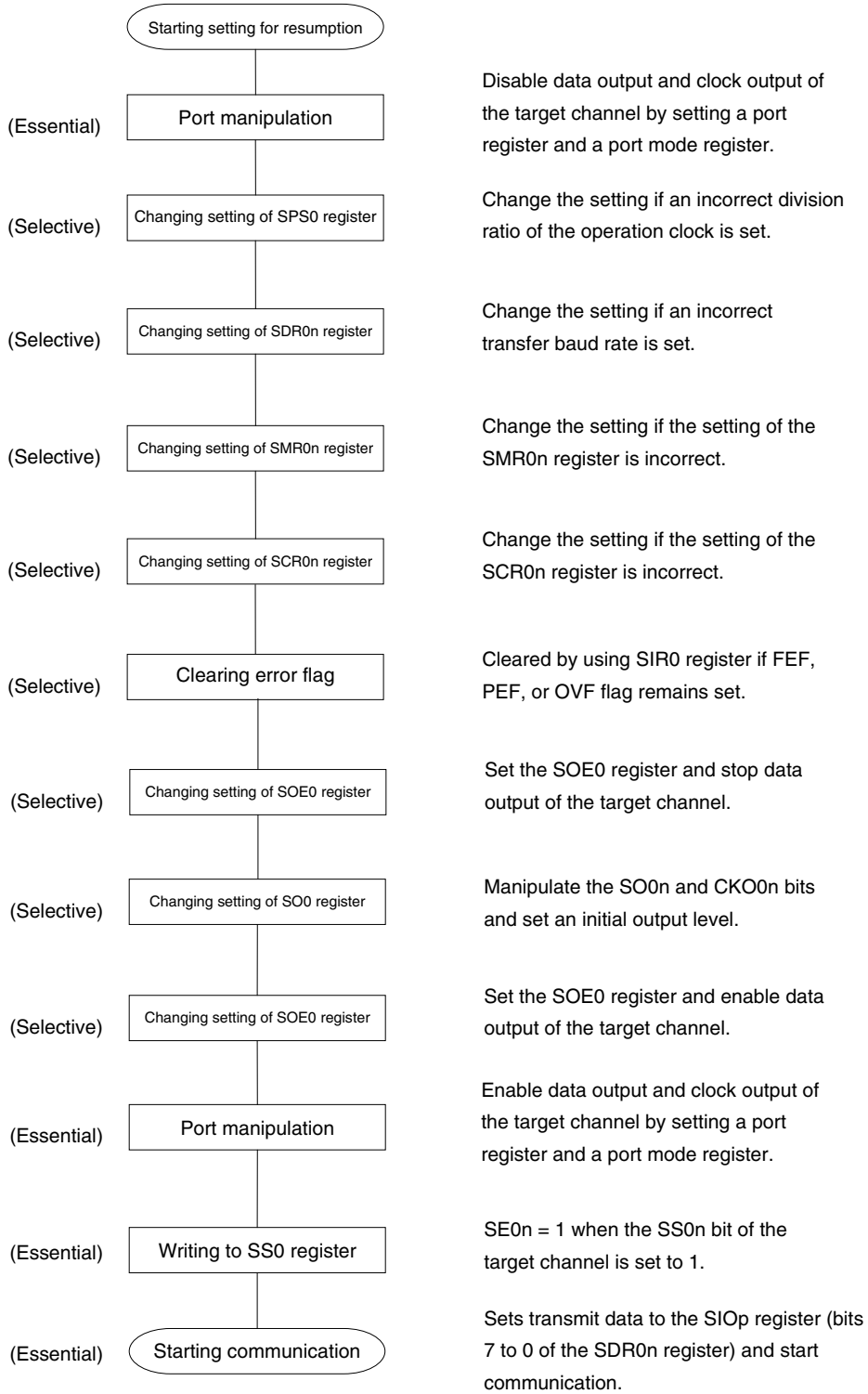
Figure 11-26. Procedure for Stopping Master Transmission



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO0 register (see **Figure 11-27 Procedure for Resuming Master Transmission**).

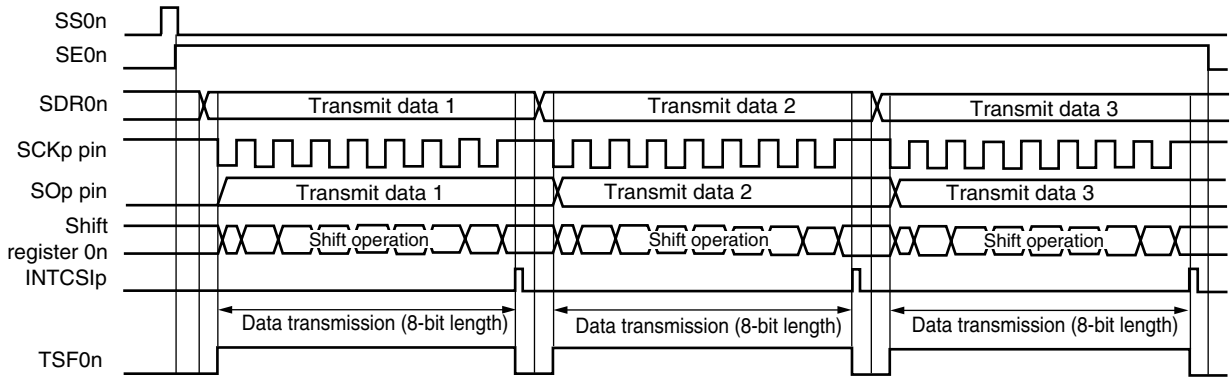
2. p: CSI number (p = 00, 10)

Figure 11-27. Procedure for Resuming Master Transmission



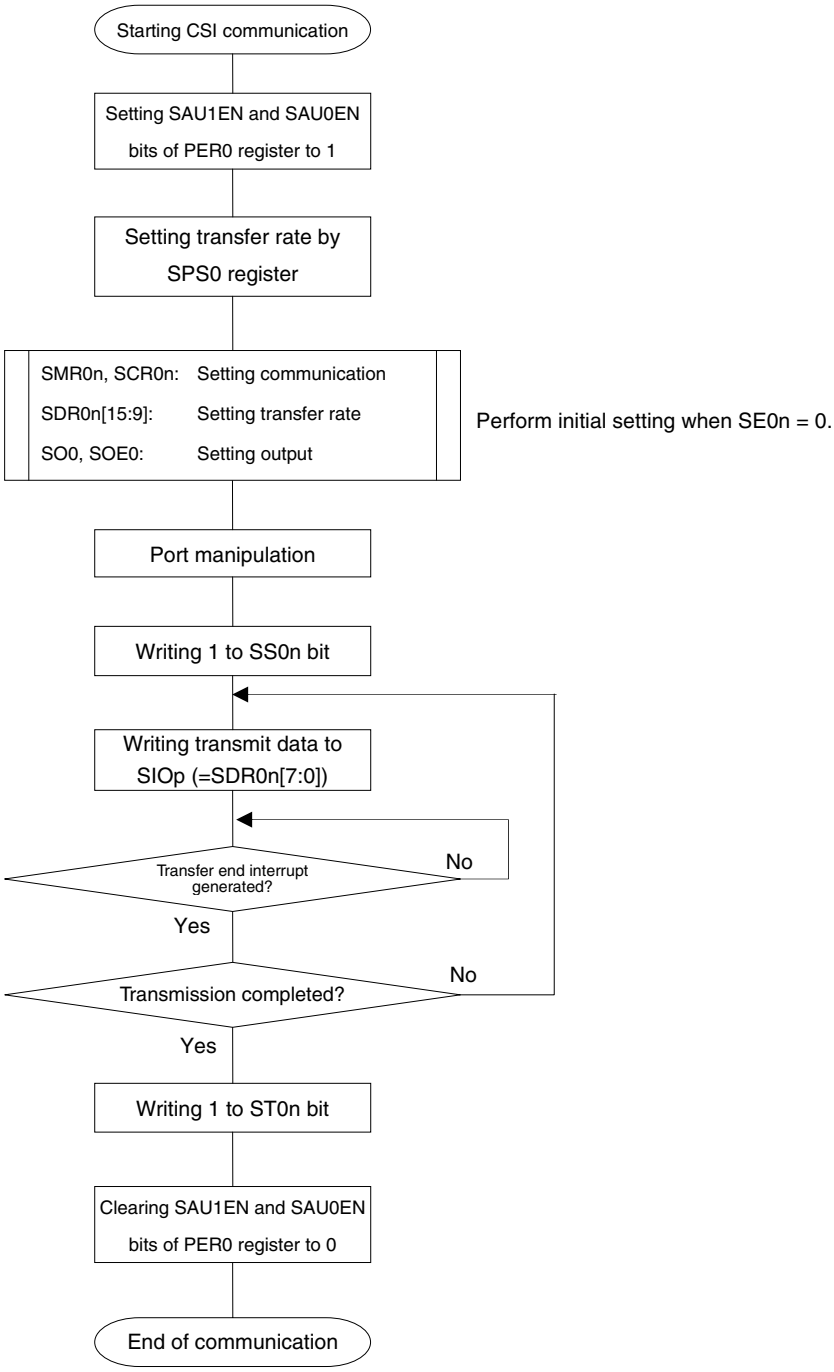
(3) Processing flow (in single-transmission mode)

Figure 11-28. Timing Chart of Master Transmission (in Single-Transmission Mode)



Remark n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)

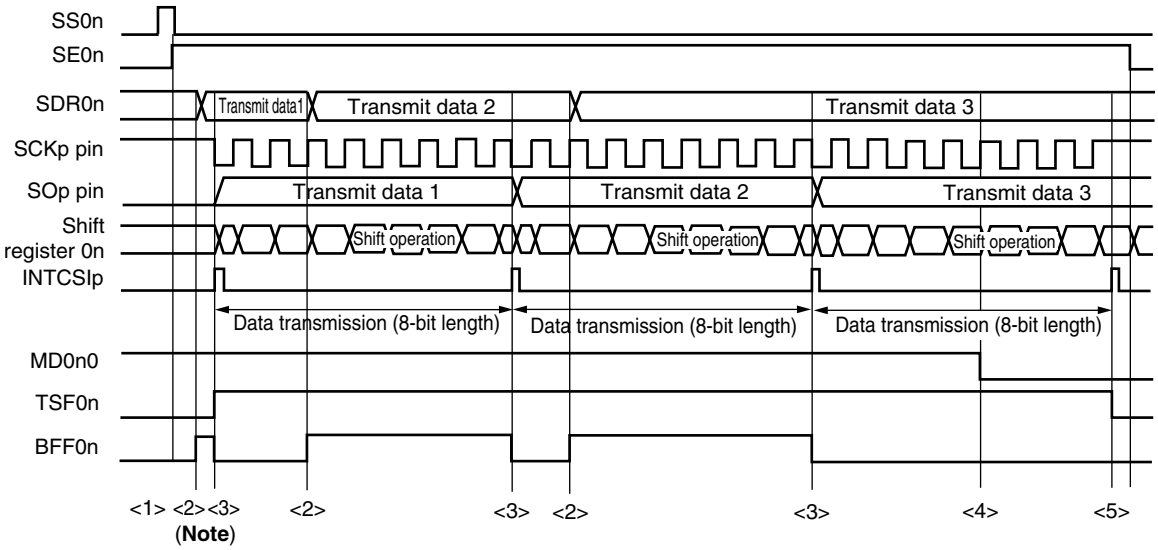
Figure 11-29. Flowchart of Master Transmission (in Single-Transmission Mode)



Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 11-30. Timing Chart of Master Transmission (in Continuous Transmission Mode)

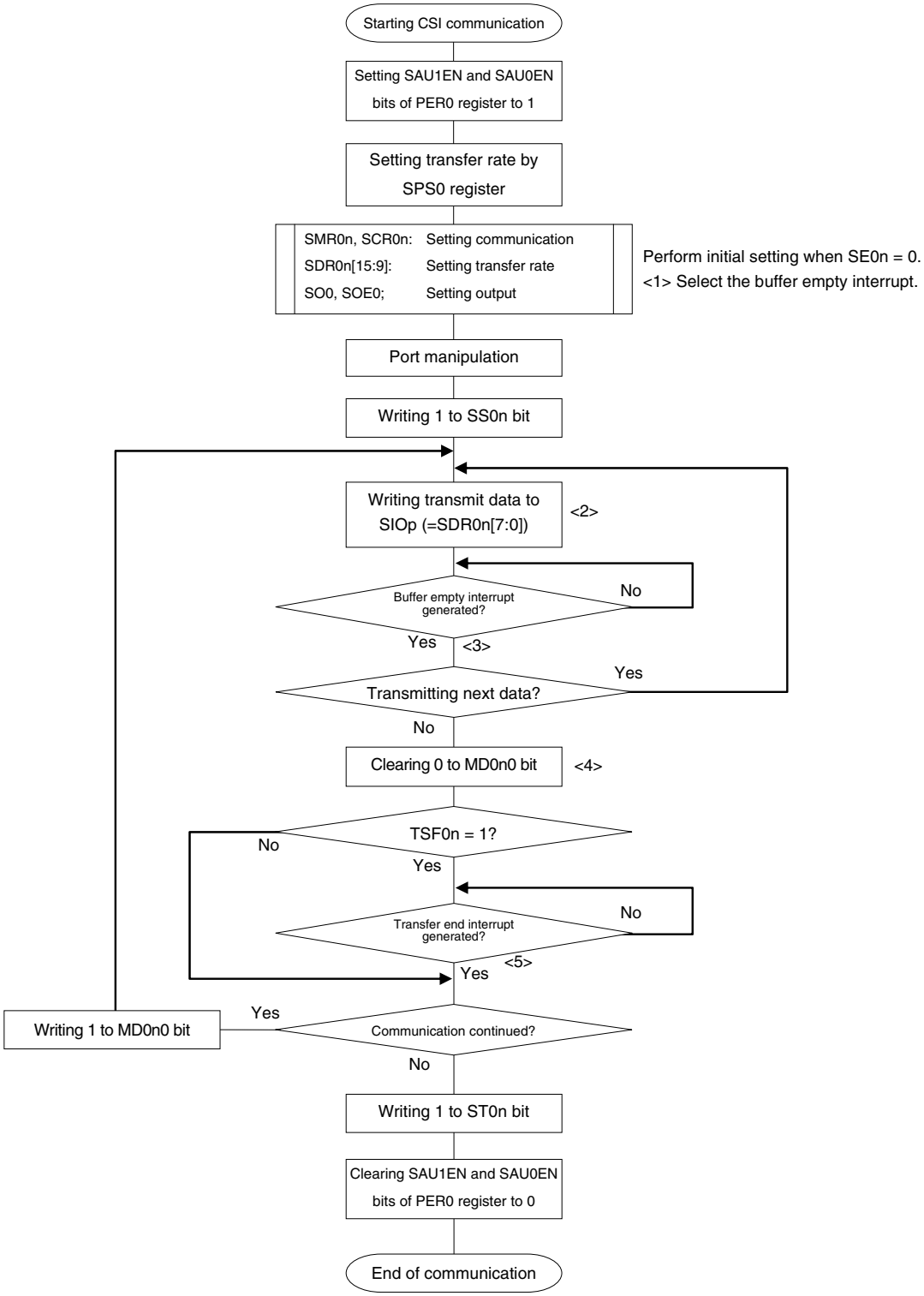


Note When transmit data is written to the SDR0n register while BFF0n = 1, the transmit data is overwritten.

Caution The MD0n0 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 n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)

Figure 11-31. Flowchart of Master Transmission (in Continuous Transmission Mode)



Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Remark <1> to <5> in the figure correspond to <1> to <5> in **Figure 11-30 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

11.5.2 Master reception

Master reception is that the 78K0R/KE3 outputs a transfer clock and receives data from other device.

3-Wire Serial I/O	CSI00	CSI10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCK00, SI00	SCK10, SI10
Interrupt	INTCSI00	INTCSI10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVF0n) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz] Note f_{CLK}: System clock frequency	
Data phase	Selectable by DAP0n bit • DAP0n = 0: Data input starts from the start of the operation of the serial clock. • DAP0n = 1: Data input starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by CKP0n bit • CKP0n = 0: Forward • CKP0n = 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 27 ELECTRICAL SPECIFICATIONS**).

(1) Register setting

Figure 11-32. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI10)

(a) Serial output register 0 (SO0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 0/1	1	CKO00 0/1	0	0	0	0	1	SO02 ×	1	SO00 ×

Communication starts when these bits are 1 if the data phase is forward (CKP0n = 0). If the phase is reversed (CKP0n = 1), communication starts when these bits are 0.

(b) Serial output enable register 0 (SOE0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	0	SOE00 0/1

Set these bits to 0 when using them for CSI master reception

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 0/1

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 0	0	0	0	0	0	STS0n 0	0	SIS0n0 0	1	0	0	MD0n2 0	MD0n1 0	MD0n0 0

Operation mode of channel n
0: Transfer end interrupt

(e) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n 0	RXE0n 1	DAP0n 0/1	CKP0n 0/1	0	EOC0n 0	PTC0n1 0	PTC0n0 0	DIR0n 0/1	0	SLC0n1 0	SLC0n0 0	0	DLS0n2 1	DLS0n1 1	DLS0n0 0/1

(f) Serial data register 0n (SDR0n) (lower 8 bits: SIOp)

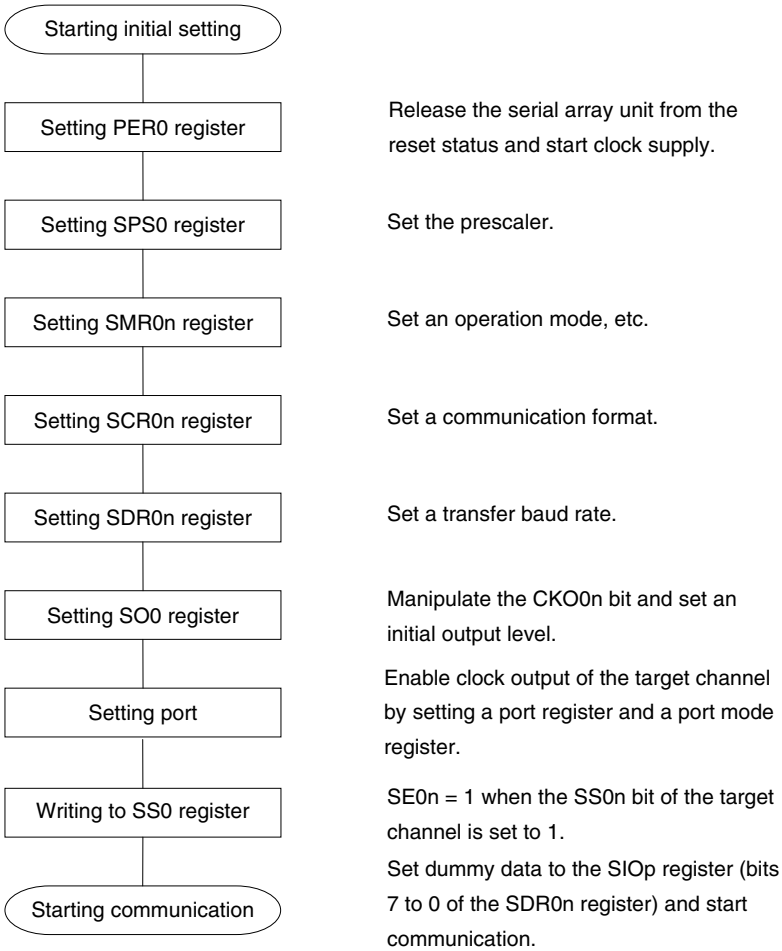
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	Baud rate setting							0	Receive data register (Write FFH as dummy data.)							

SIOp

Remark n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)
□: 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

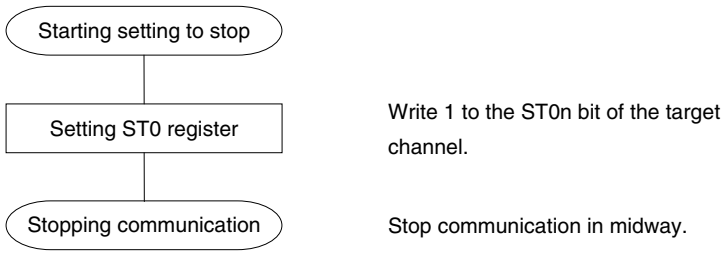
(2) Operation procedure

Figure 11-33. Initial Setting Procedure for Master Reception



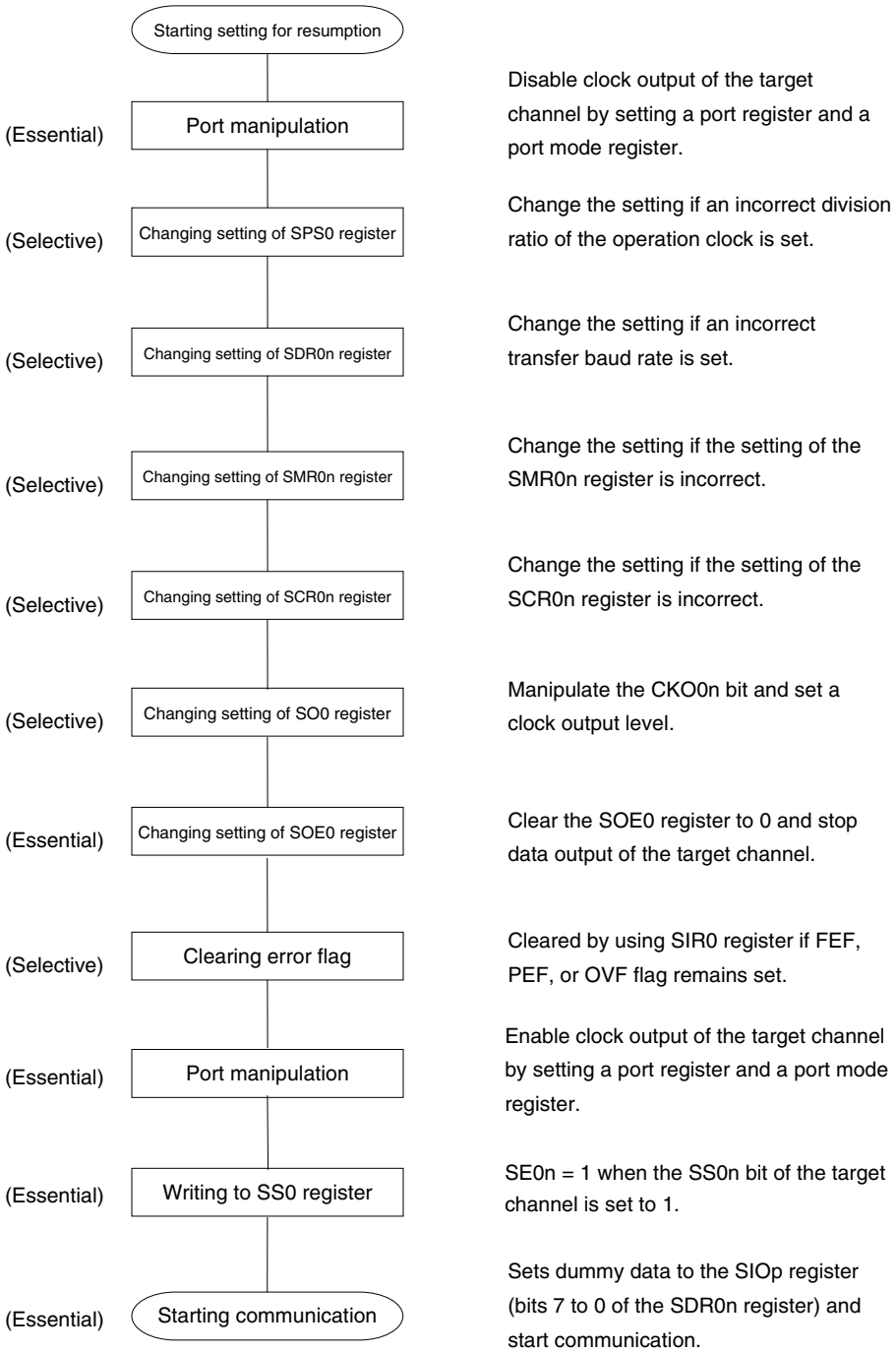
Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Figure 11-34. Procedure for Stopping Master Reception



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO0 register (see **Figure 11-35 Procedure for Resuming Master Reception**).

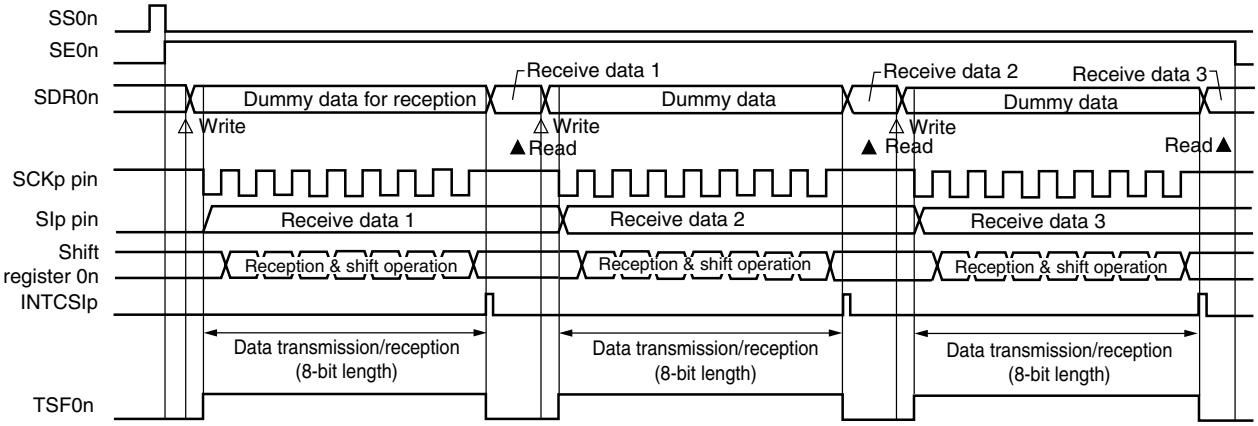
Figure 11-35. Procedure for Resuming Master Reception



(3) Processing flow (in single-reception mode)

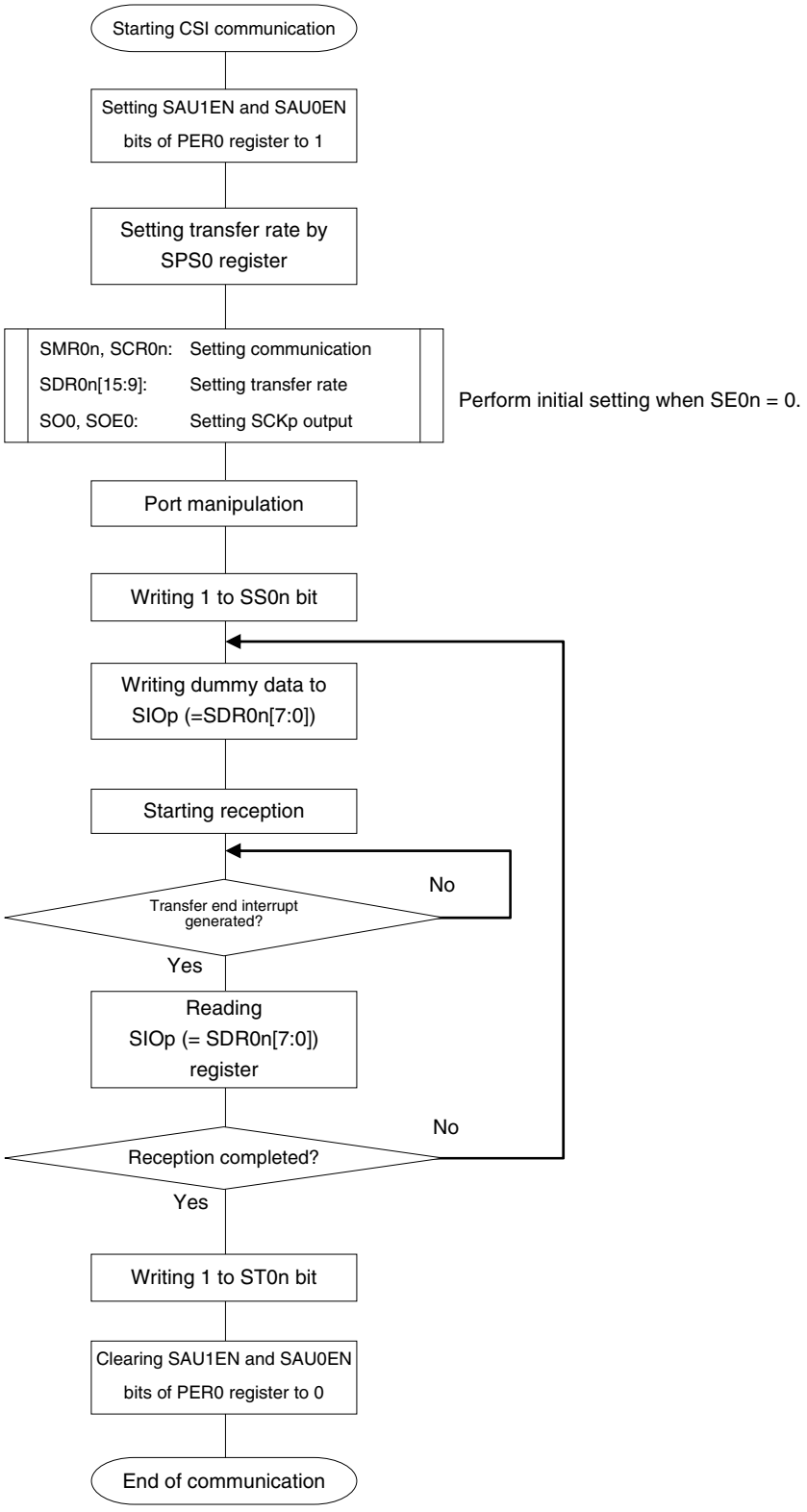
Figure 11-36. Timing Chart of Master Reception (in Single-Reception Mode)

<R>



Remark n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)

Figure 11-37. Flowchart of Master Reception (in Single-Reception Mode)



Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

11.5.3 Master transmission/reception

Master transmission/reception is that the 78K0R/KE3 outputs a transfer clock and transmits/receives data to/from other device.

3-Wire Serial I/O	CSI00	CSI10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCK00, SI00, SO00	SCK10, SI10, SO10
Interrupt	INTCSI00	INTCSI10
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVF0n) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz]Note f_{CLK}: System clock frequency	
Data phase	Selectable by DAP0n bit • DAP0n = 0: Data output starts at the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by CKP0n bit • CKP0n = 0: Forward • CKP0n = 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 27 ELECTRICAL SPECIFICATIONS**).

(1) Register setting

Figure 11-38. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI10)

(a) Serial output register 0 (SO0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 0/1	1	CKO00 0/1	0	0	0	0	1	SO02 0/1	1	SO00 0/1

Communication starts when these bits are 1 if the data phase is forward (CKP0n = 0). If the phase is reversed (CKP0n = 1), communication starts when these bits are 0.

(b) Serial output enable register 0 (SOE0) … 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
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	0	SOE00 0/1

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 0/1

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 0	0	0	0	0	0	STS0n 0	0	SIS0n0 0	1	0	0	MD0n2 0	MD0n1 0	MD0n0 0/1

Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n 1	RXE0n 1	DAP0n 0/1	CKP0n 0/1	0	EOC0n 0	PTC0n1 0	PTC0n0 0	DIR0n 0/1	0	SLC0n1 0	SLC0n0 0	0	DLS0n2 1	DLS0n1 1	DLS0n0 0/1

(f) Serial data register 0n (SDR0n) (lower 8 bits: SIOp)

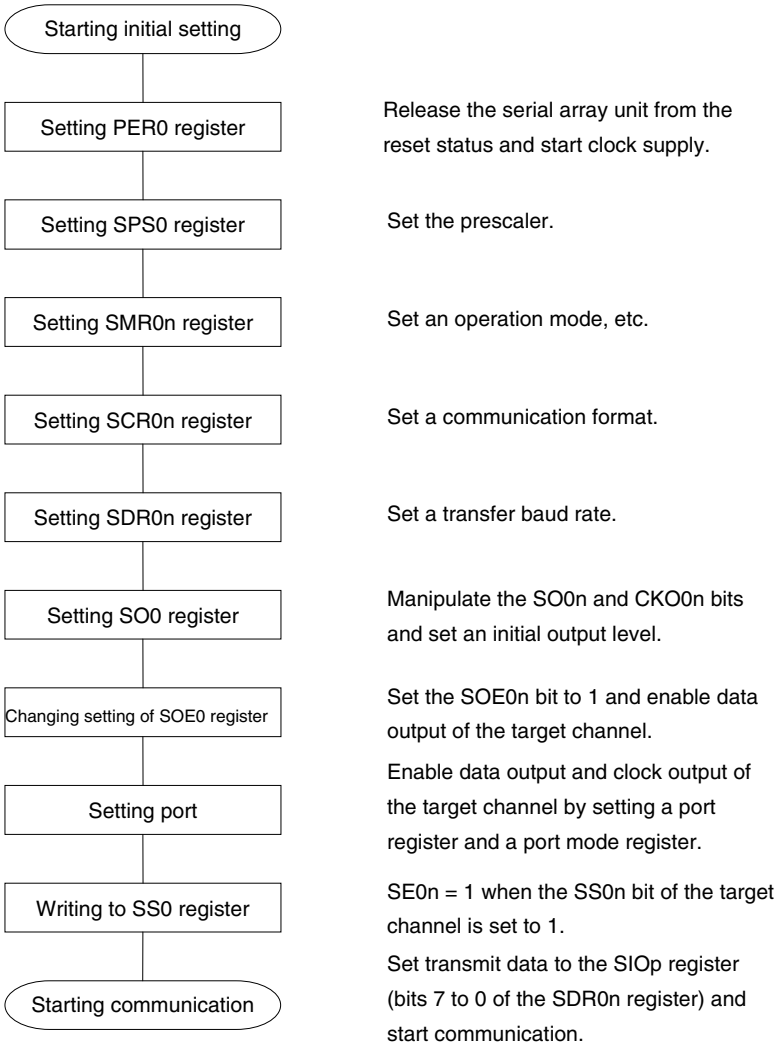
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	Baud rate setting							0	Transmit data setting/receive data register							

SIOp

Remark n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)
□: 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

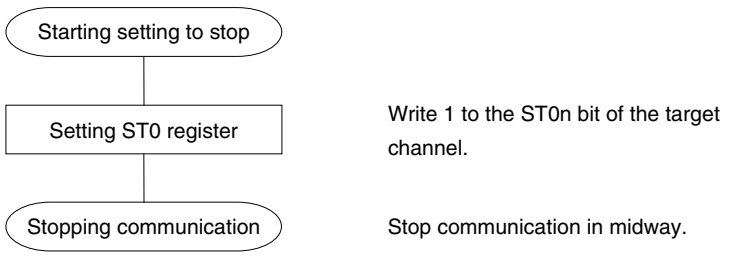
(2) Operation procedure

Figure 11-39. Initial Setting Procedure for Master Transmission/Reception



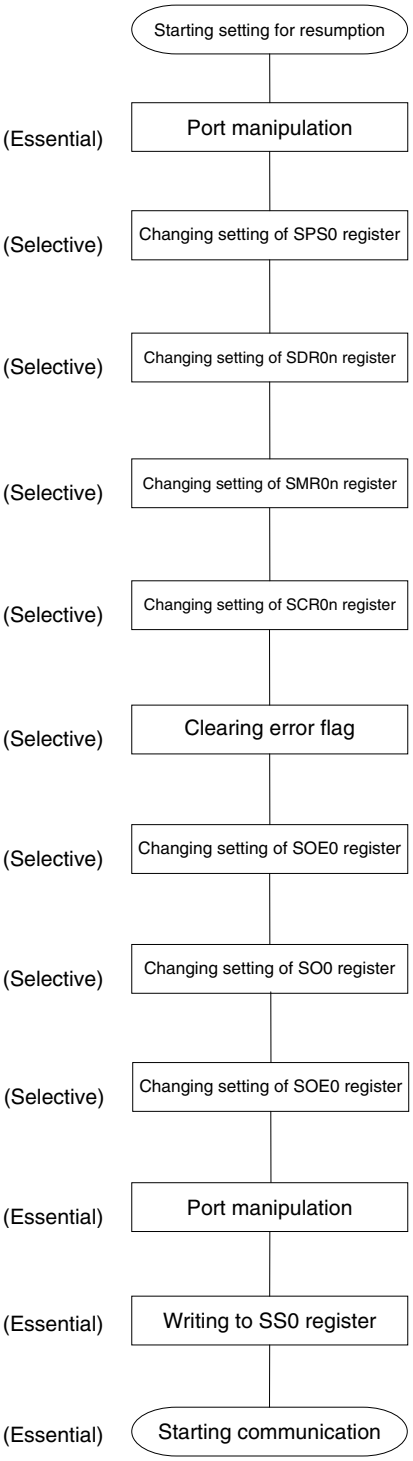
Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Figure 11-40. Procedure for Stopping Master Transmission/Reception



Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO0 register (see **Figure 11-41 Procedure for Resuming Master Transmission/Reception**).

Figure 11-41. Procedure for Resuming Master Transmission/Reception



Disable data output and clock output of the target channel by setting a port register and a port mode register.

Change the setting if an incorrect division ratio of the operation clock is set.

Change the setting if an incorrect transfer baud rate is set.

Change the setting if the setting of the SMR0n register is incorrect.

Change the setting if the setting of the SCR0n register is incorrect.

Cleared by using SIR0 register if FEF, PEF, or OVF flag remains set.

Set the SOE0 register and stop data output of the target channel.

Manipulate the SO0n and CKO0n bits and set an initial output level.

Set the SOE0 register and enable data output of the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

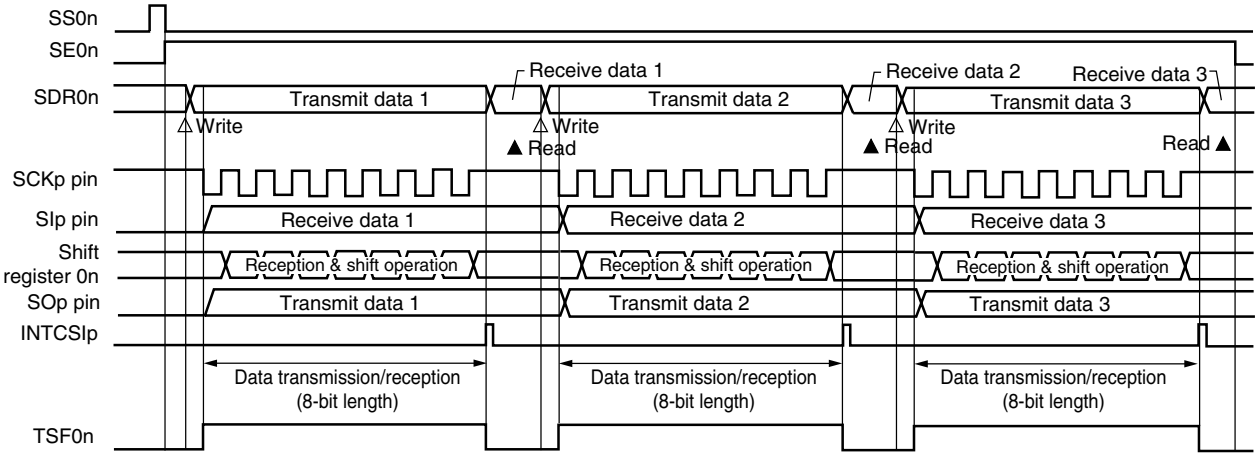
SE0n = 1 when the SS0n bit of the target channel is set to 1.

Sets transmit data to the SIOp register (bits 7 to 0 of the SDR0n register) and start communication.

(3) Processing flow (in single-transmission/reception mode)

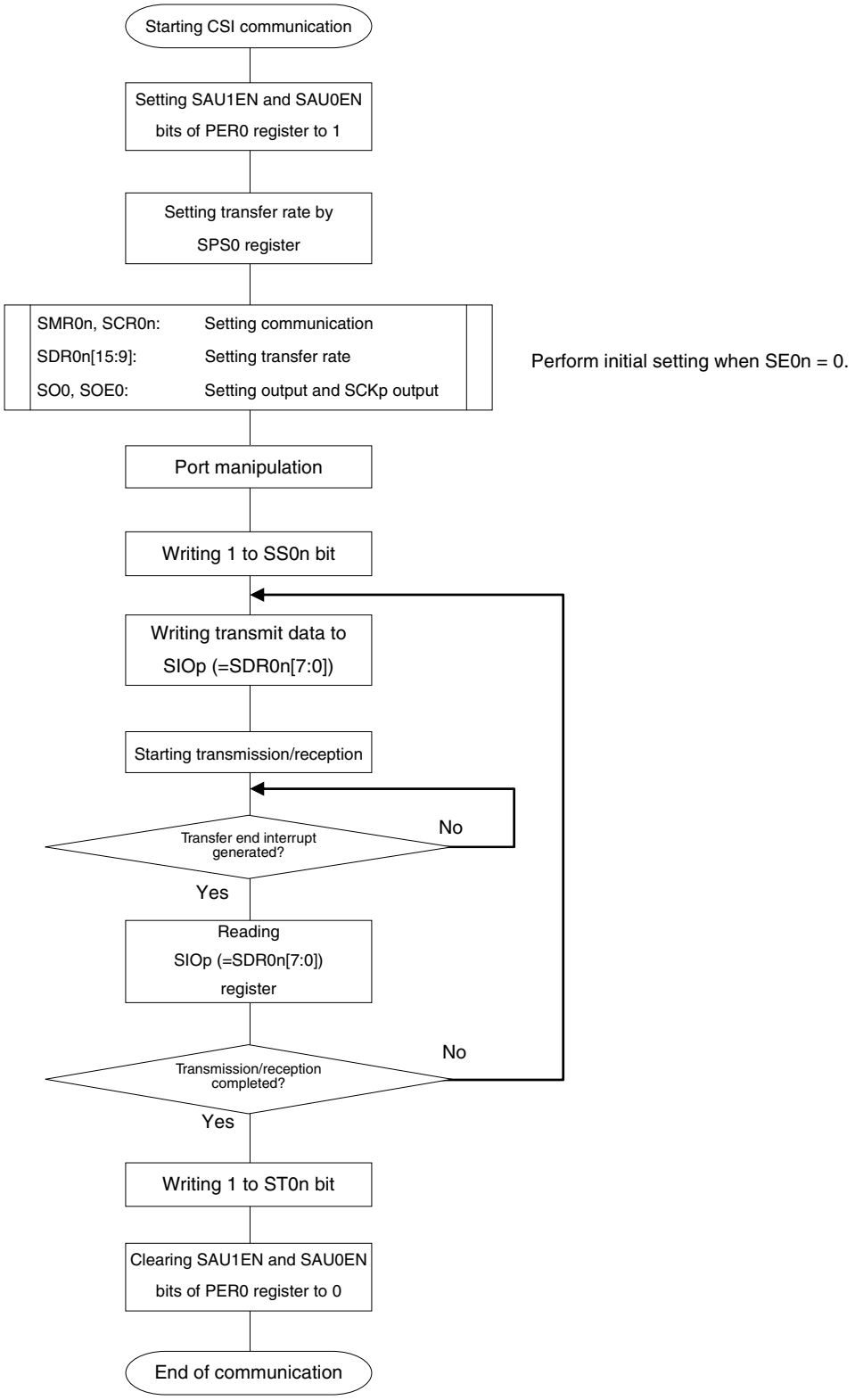
Figure 11-42. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode)

<R>



Remark n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)

Figure 11-43. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)

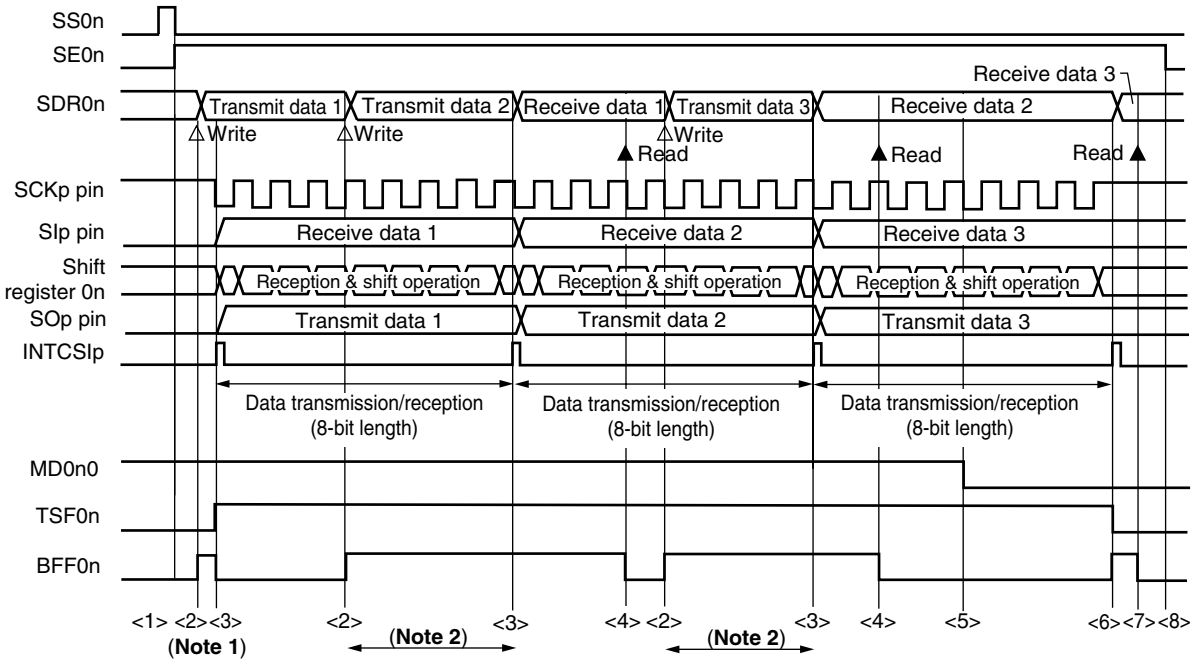


Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-44. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)

<R>



Notes 1. When transmit data is written to the SDR0n register while BFF0n = 1, the transmit data is overwritten.

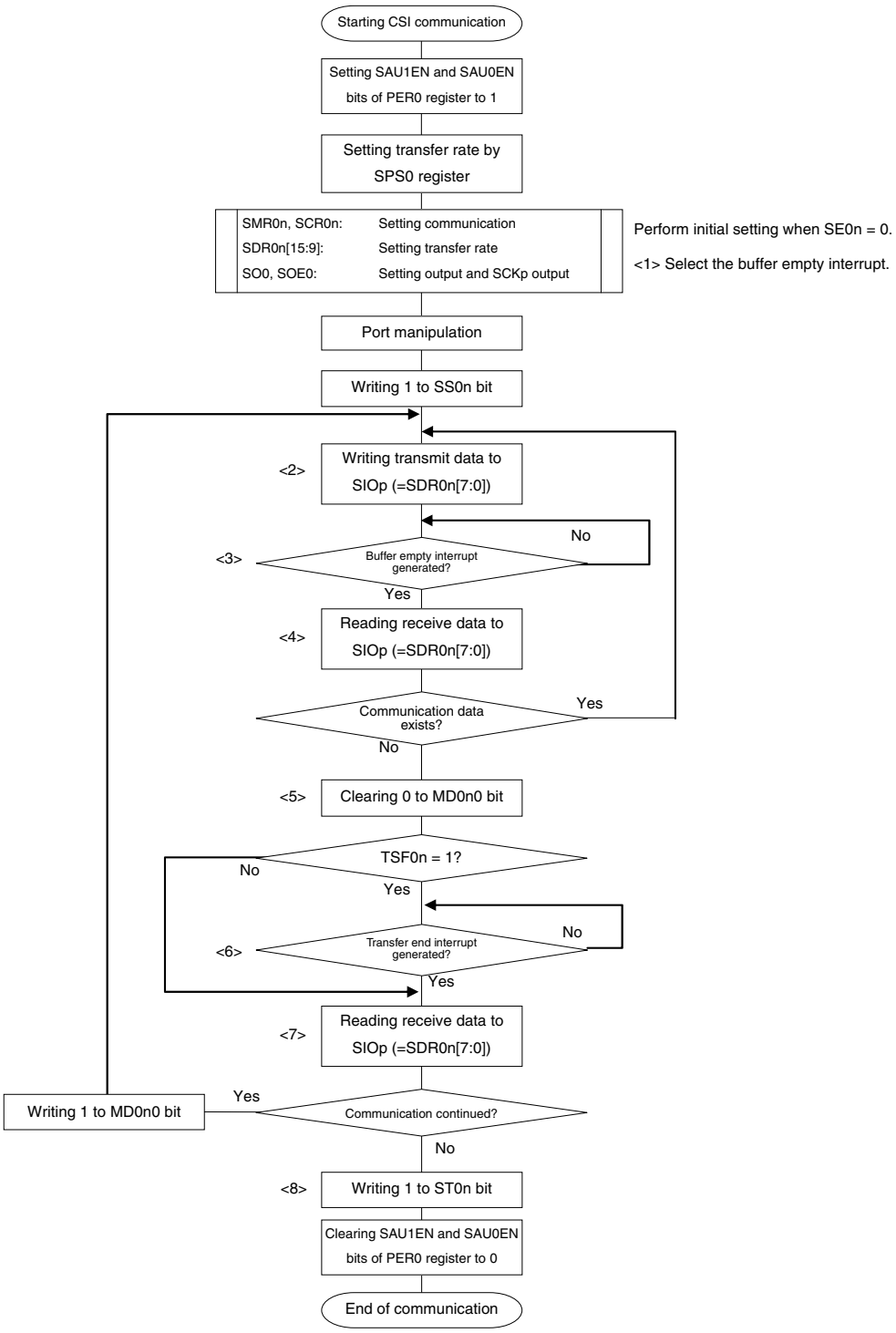
2. The transmit data can be read by reading the SDR0n register during this period. At this time, the transfer operation is not affected.

Caution The MD0n0 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 11-45 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)

Figure 11-45. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



<R>

Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-44 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

11.5.4 Slave transmission

Slave transmission is that the 78K0R/KE3 transmits data to another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCK00, SO00	SCK10, SO10
Interrupt	INTCSI00	INTCSI10
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVF0n) only	
Transfer data length	7 or 8 bits	
Transfer rate	The smaller of f_{CLK}/6 [MHz] and f_{MCK}/2 [MHz] is the maximum transfer rate Notes 1, 2.	
Data phase	Selectable by DAP0n bit • DAP0n = 0: Data output starts from the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by CKP0n bit • CKP0n = 0: Forward • CKP0n = 1: Reverse	
Data direction	MSB or LSB first	

Notes 1. Because the external serial clock input to pins SCK00, SCK01, SCK10, and SCK20 is sampled internally and used, the fastest baud rate is the smaller of f_{CLK}/6 [MHz] and f_{MCK}/2 [MHz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 27 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-46. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI10)

(a) Serial output register 0 (SO0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 ×	1	CKO00 ×	0	0	0	0	1	SO02 0/1	1	SO00 0/1

(b) Serial output enable register 0 (SOE0) … 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
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	0	SOE00 0/1

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 0/1

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 1	0	0	0	0	0	STS0n 0	0	SIS0n0 0	1	0	0	MD0n2 0	MD0n1 0	MD0n0 0/1

Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n 1	RXE0n 0	DAP0n 0/1	CKP0n 0/1	0	EOC0n 0	PTC0n1 0	PTC0n0 0	DIR0n 0/1	0	SLC0n1 0	SLC0n0 0	0	DLS0n2 1	DLS0n1 1	DLS0n0 0/1

(f) Serial data register 0n (SDR0n) (lower 8 bits: SIOp)

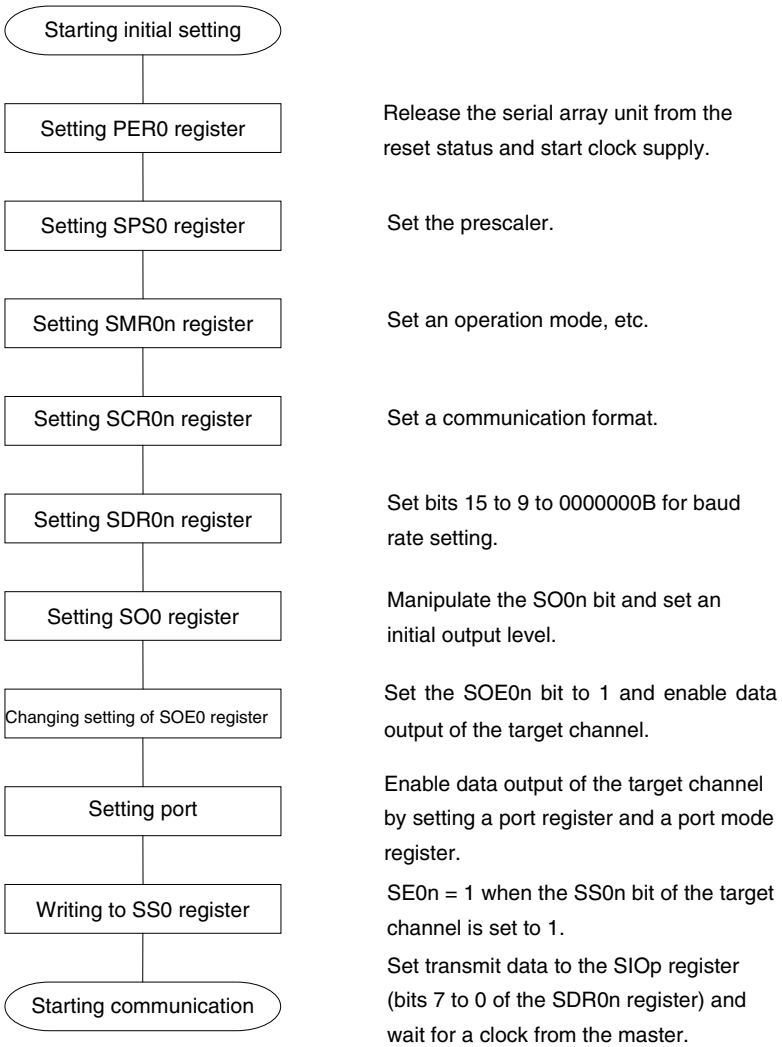
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	Baud rate setting							0	Transmit data setting							

SIOp

Remark n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)
□: 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

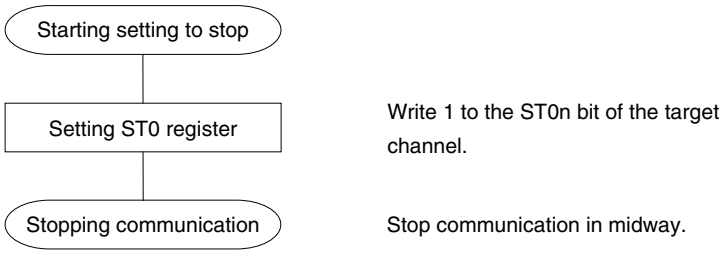
(2) Operation procedure

Figure 11-47. Initial Setting Procedure for Slave Transmission



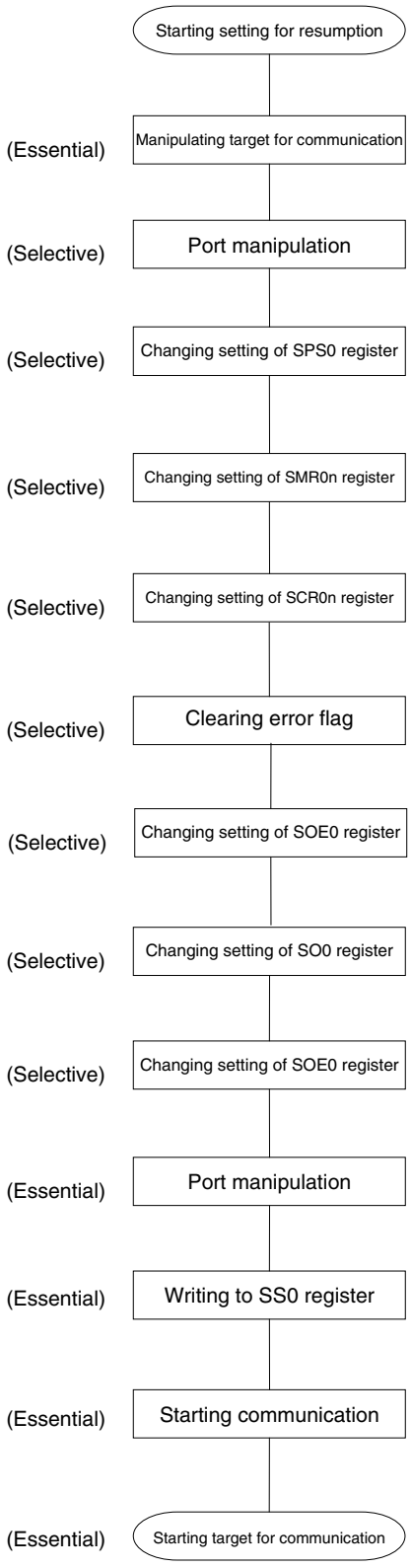
Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Figure 11-48. Procedure for Stopping Slave Transmission



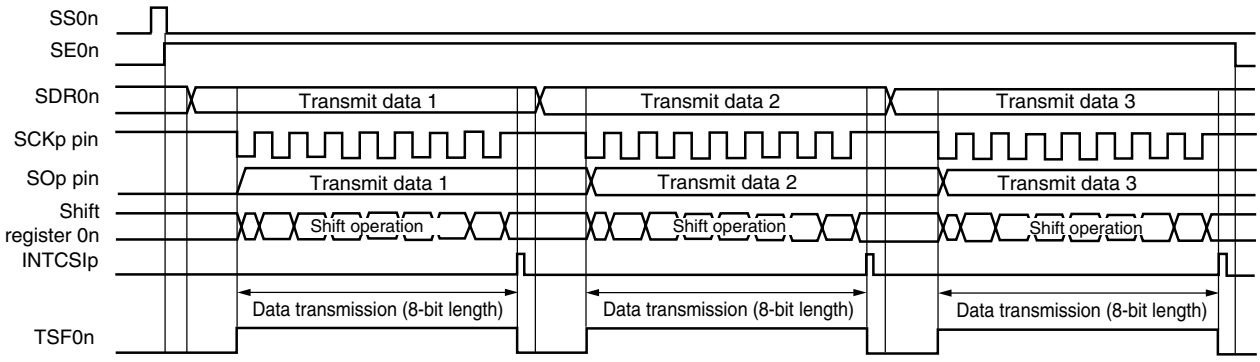
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO0 register (see **Figure 11-49 Procedure for Resuming Slave Transmission**).

Figure 11-49. Procedure for Resuming Slave Transmission



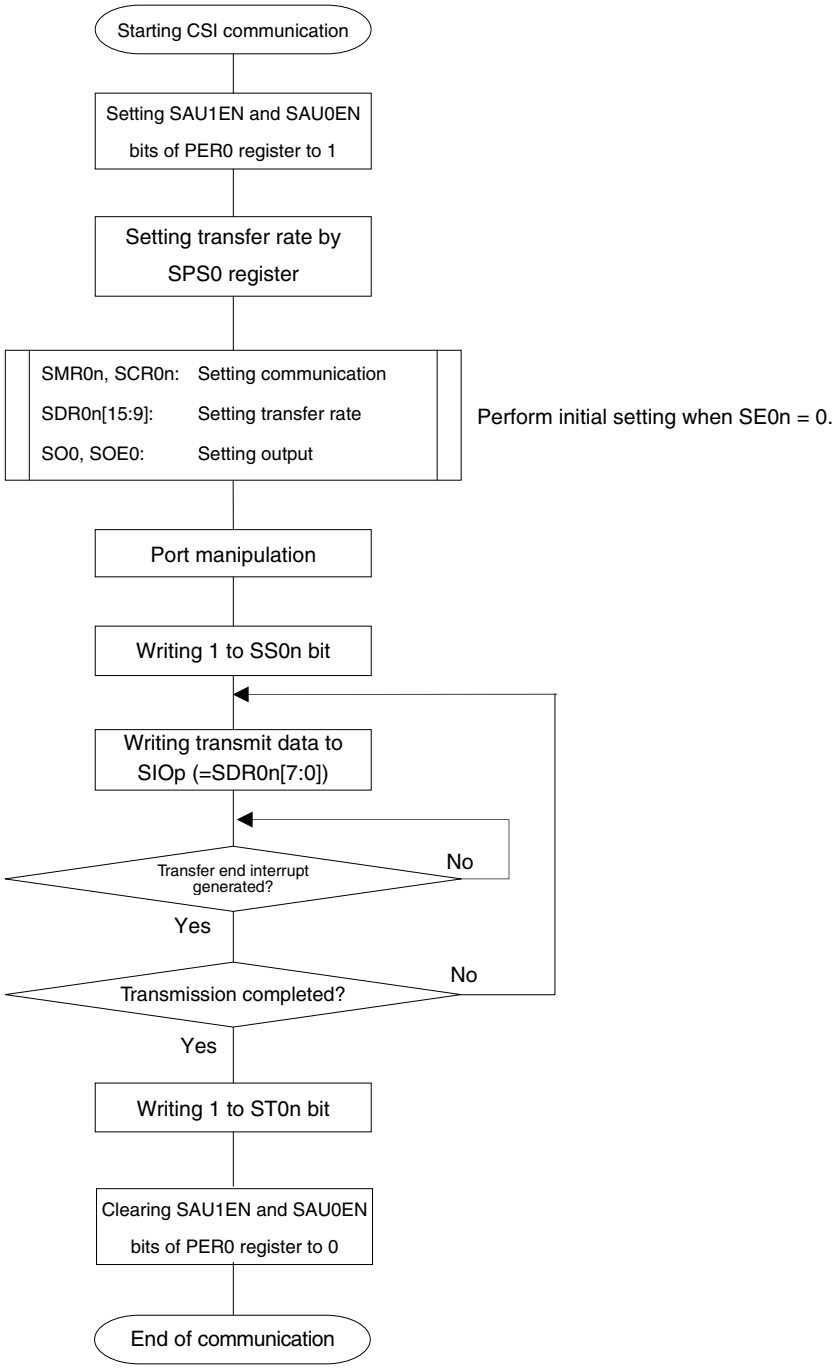
(3) Processing flow (in single-transmission mode)

<R> **Figure 11-50. Timing Chart of Slave Transmission (in Single-Transmission Mode)**



Remark n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)

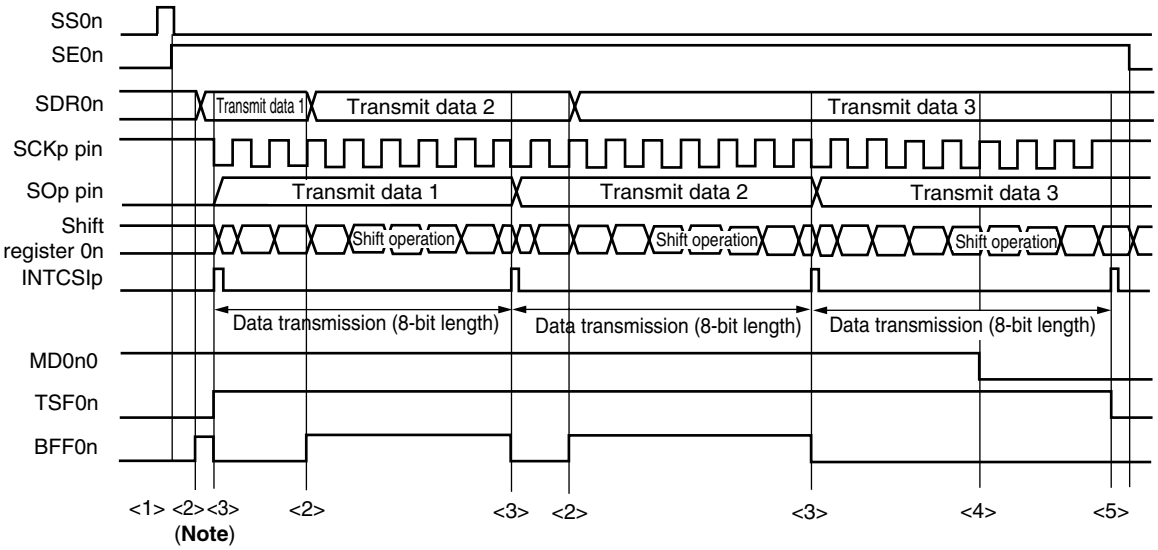
Figure 11-51. Flowchart of Slave Transmission (in Single-Transmission Mode)



Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

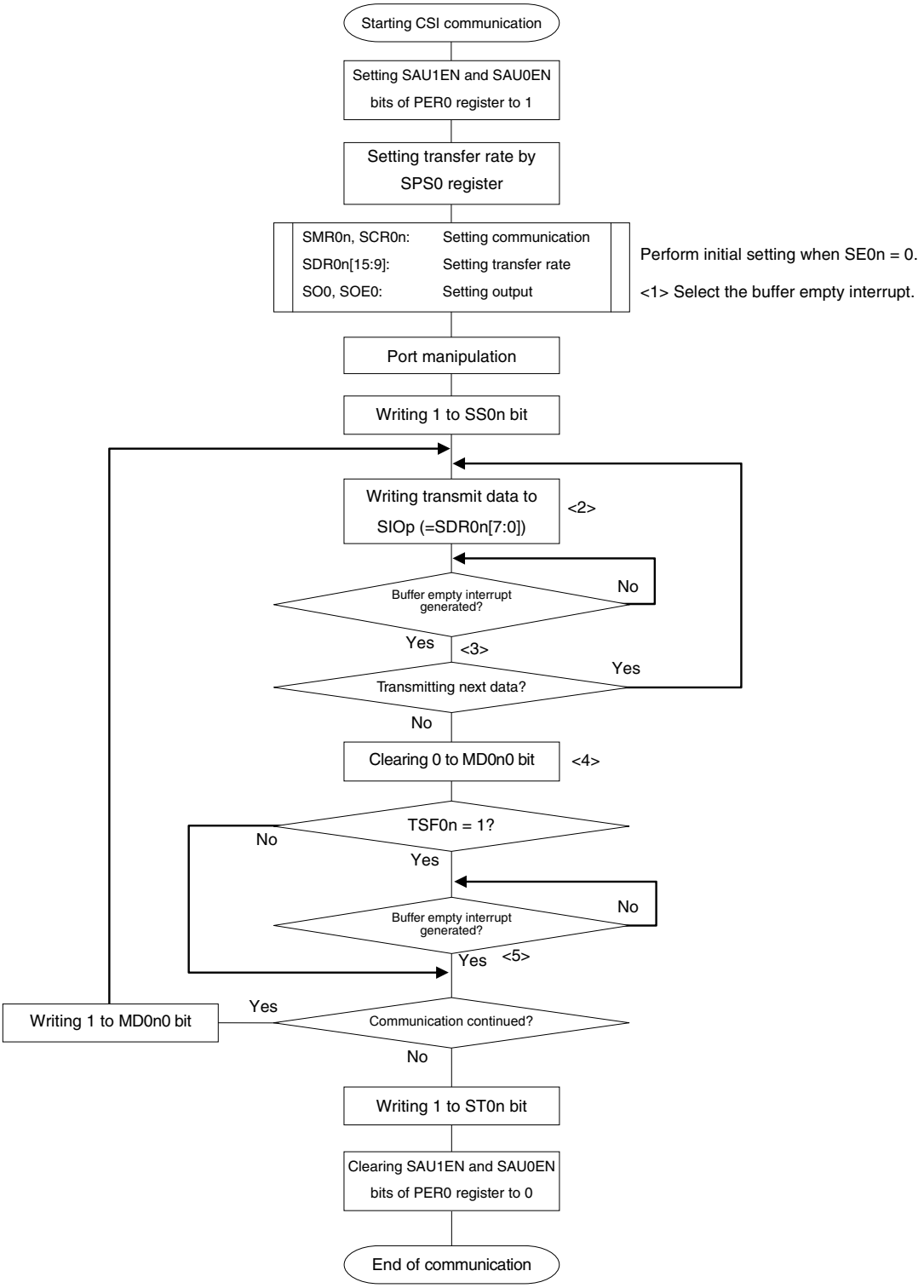
Figure 11-52. Timing Chart of Slave Transmission (in Continuous Transmission Mode)



Note When transmit data is written to the SDR0n register while BFF0n = 1, the transmit data is overwritten.

Caution The MD0n0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Figure 11-53. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Remark <1> to <5> in the figure correspond to <1> to <5> in **Figure 11-52 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

11.5.5 Slave reception

Slave reception is that the 78K0R/KE3 receives data from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCK00, SI00	SCK10, SI10
Interrupt	INTCSI00	INTCSI10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVF0n) only	
Transfer data length	7 or 8 bits	
Transfer rate	The smaller of f_{CLK}/6 [MHz] and f_{MCK}/2 [MHz] is the maximum transfer rate[Notes 1, 2].	
Data phase	Selectable by DAP0n bit • DAP0n = 0: Data input starts from the start of the operation of the serial clock. • DAP0n = 1: Data input starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by CKP0n bit • CKP0n = 0: Forward • CKP0n = 1: Reverse	
Data direction	MSB or LSB first	

Notes 1. Because the external serial clock input to pins SCK00, SCK01, SCK10, and SCK20 is sampled internally and used, the fastest baud rate is the smaller of f_{CLK}/6 [MHz] and f_{MCK}/2 [MHz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 27 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-54. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI10)

(a) Serial output register 0 (SO0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 ×	1	CKO00 ×	0	0	0	0	1	SO02 ×	1	SO00 ×

(b) Serial output enable register 0 (SOE0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	0	SOE00 0/1

Set these bits to 0 when using them for CSI slave reception

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 0/1

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 1	0	0	0	0	0	STS0n 0	0	SIS0n0 0	1	0	0	MD0n2 0	MD0n1 0	MD0n0 0

Operation mode of channel n
0: Transfer end interrupt

(e) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n 0	RXE0n 1	DAP0n 0/1	CKP0n 0/1	0	EOC0n 0	PTC0n1 0	PTC0n0 0	DIR0n 0/1	0	SLC0n1 0	SLC0n0 0	0	DLS0n2 1	DLS0n1 1	DLS0n0 0/1

(f) Serial data register 0n (SDR0n) (lower 8 bits: SIOp)

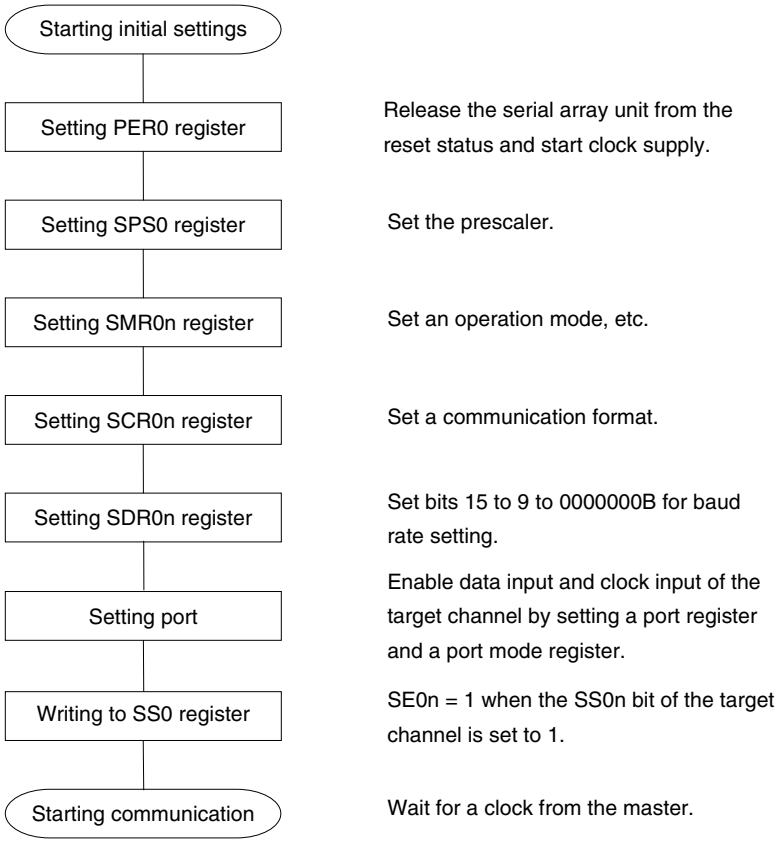
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	0000000 (baud rate setting)							0	Receive data register							

SIOp

Remark n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)
□: 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 11-55. Initial Setting Procedure for Slave Reception



Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Figure 11-56. Procedure for Stopping Slave Reception

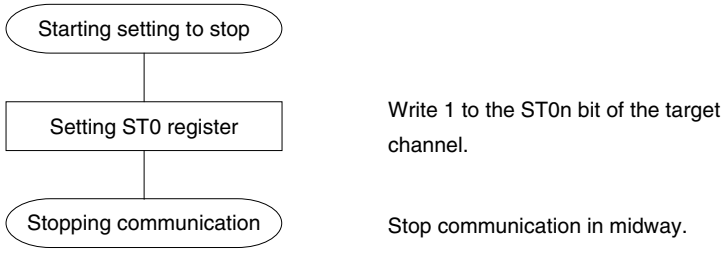
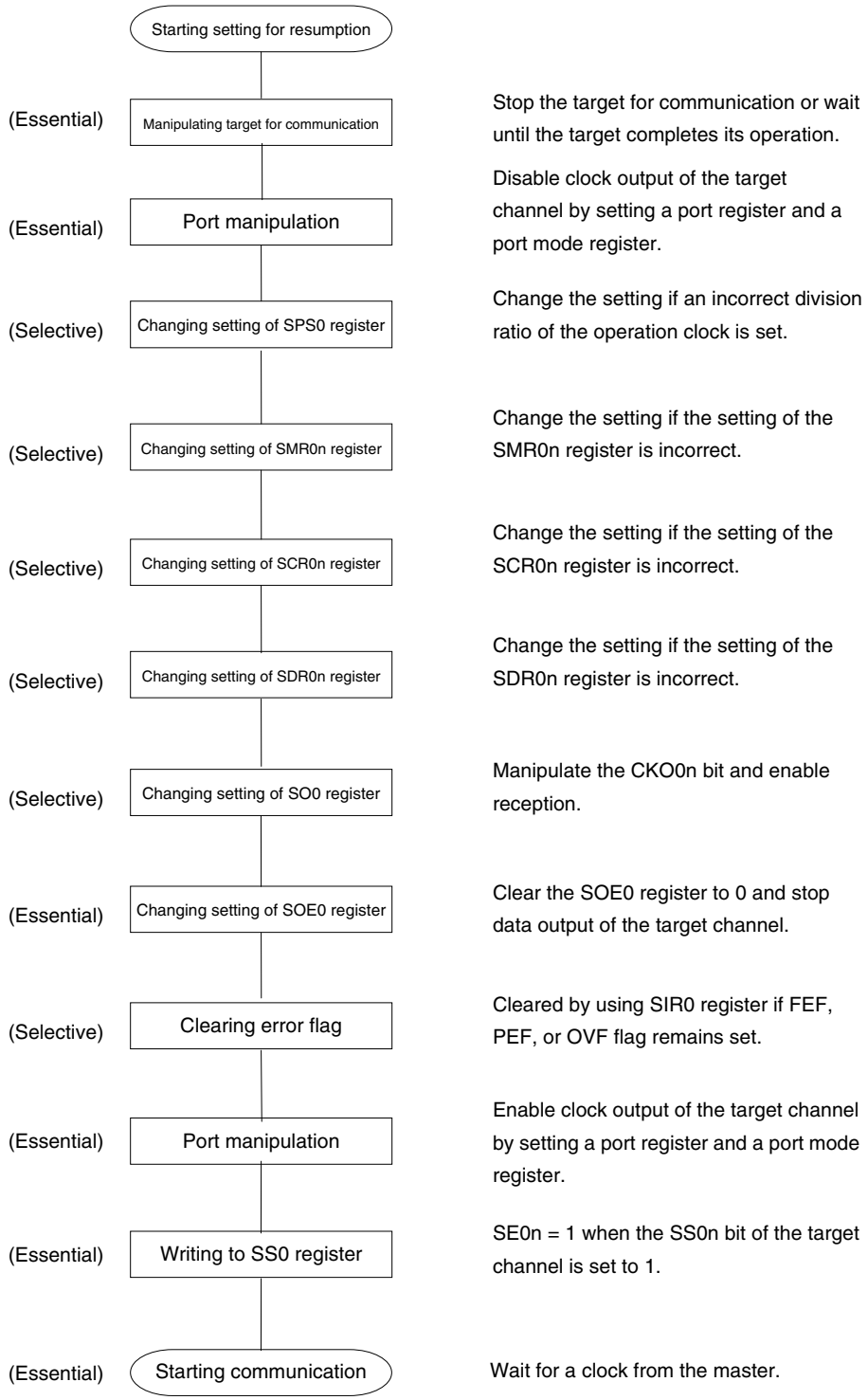
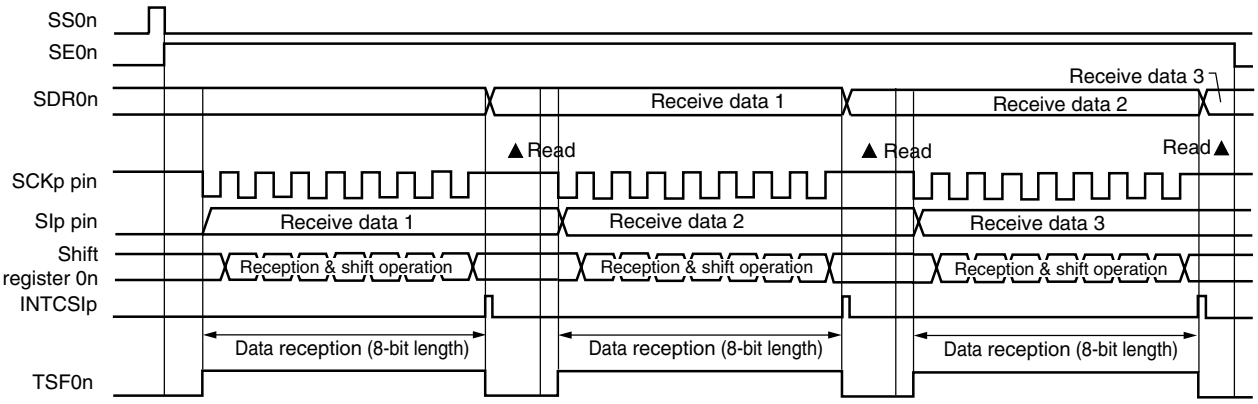


Figure 11-57. Procedure for Resuming Slave Reception



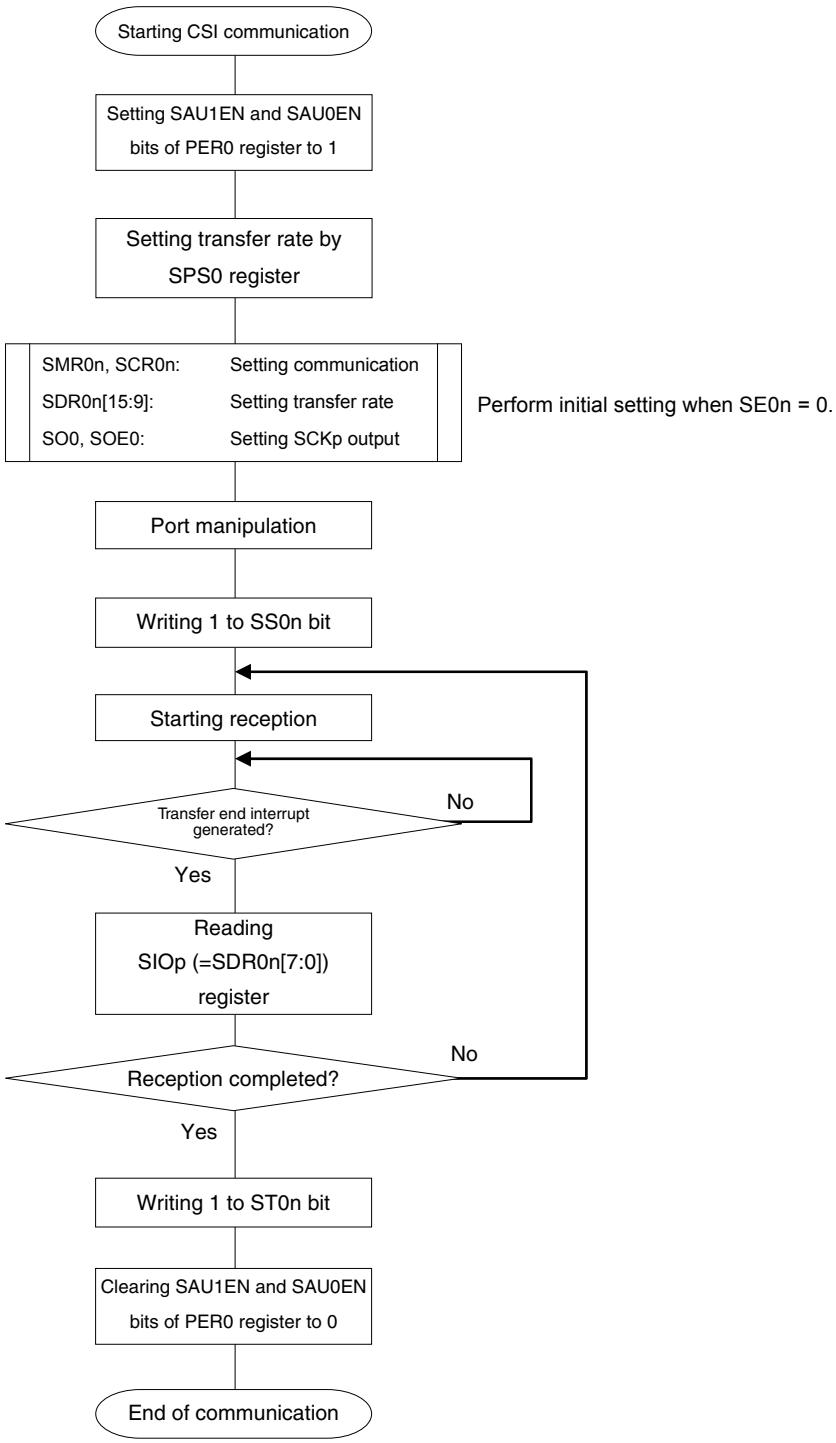
(3) Processing flow (in single-reception mode)

<R> **Figure 11-58. Timing Chart of Slave Reception (in Single-Reception Mode)**



Remark n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)

Figure 11-59. Flowchart of Slave Reception (in Single-Reception Mode)



Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

11.5.6 Slave transmission/reception

Slave transmission/reception is that the 78K0R/KE3 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	CSI10
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	SCK00, SI00, SO00	SCK10, SI10, SO10
Interrupt	INTCSI00	INTCSI10
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVF0n) only	
Transfer data length	7 or 8 bits	
Transfer rate	The smaller of f_{CLK}/6 [MHz] and f_{MCK}/2 [MHz] is the maximum transfer rate Notes 1, 2.	
Data phase	Selectable by DAP0n bit • DAP0n = 0: Data output starts from the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by CKP0n bit • CKP0n = 0: Forward • CKP0n = 1: Reverse	
Data direction	MSB or LSB first	

Notes 1. Because the external serial clock input to pins SCK00, SCK01, SCK10, and SCK20 is sampled internally and used, the fastest baud rate is the smaller of f_{CLK}/6 [MHz] and f_{MCK}/2 [MHz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 27 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-60. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI10)

(a) Serial output register 0 (SO0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 ×	1	CKO00 ×	0	0	0	0	1	SO02 0/1	1	SO00 0/1

(b) Serial output enable register 0 (SOE0) … 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
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	0	SOE00 0/1

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 0/1

(d) Serial mode register 0n (SMR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR0n	CKS0n 0/1	CCS0n 1	0	0	0	0	0	STS0n 0	0	SIS0n0 0	1	0	0	MD0n2 0	MD0n1 0	MD0n0 0/1

Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) Serial communication operation setting register 0n (SCR0n)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR0n	TXE0n 1	RXE0n 1	DAP0n 0/1	CKP0n 0/1	0	EOC0n 0	PTC0n1 0	PTC0n0 0	DIR0n 0/1	0	SLC0n1 0	SLC0n0 0	0	DLS0n2 1	DLS0n1 1	DLS0n0 0/1

(f) Serial data register 0n (SDR0n) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR0n	0000000 (baud rate setting)							0	Transmit data setting/receive data register							

SIOp

Remark n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)

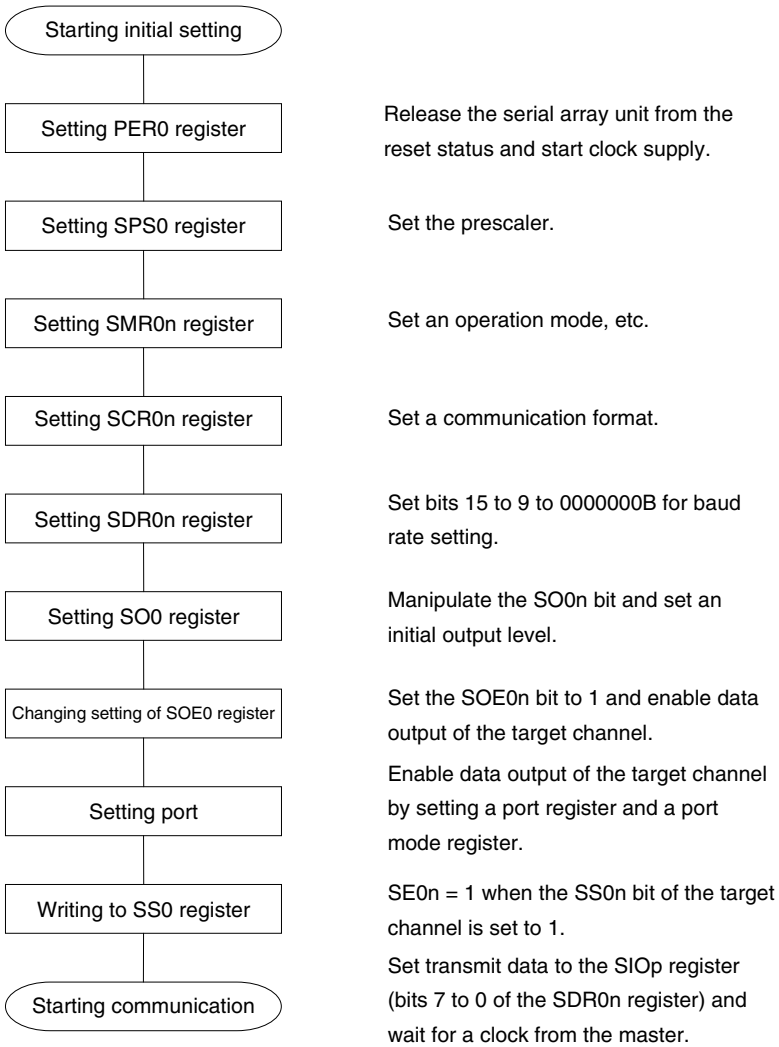
□ : Setting is fixed in the CSI slave transmission/reception mode, ▒ : Setting disabled (set to the initial value)

×: 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

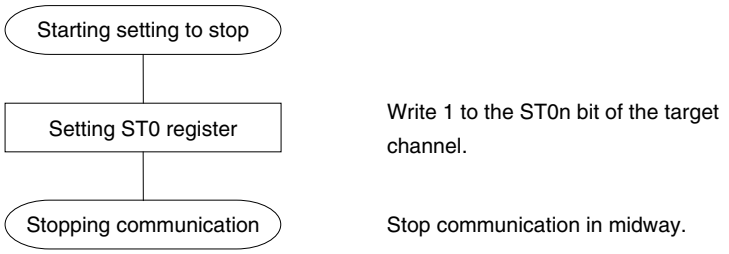
(2) Operation procedure

Figure 11-61. Initial Setting Procedure for Slave Transmission/Reception



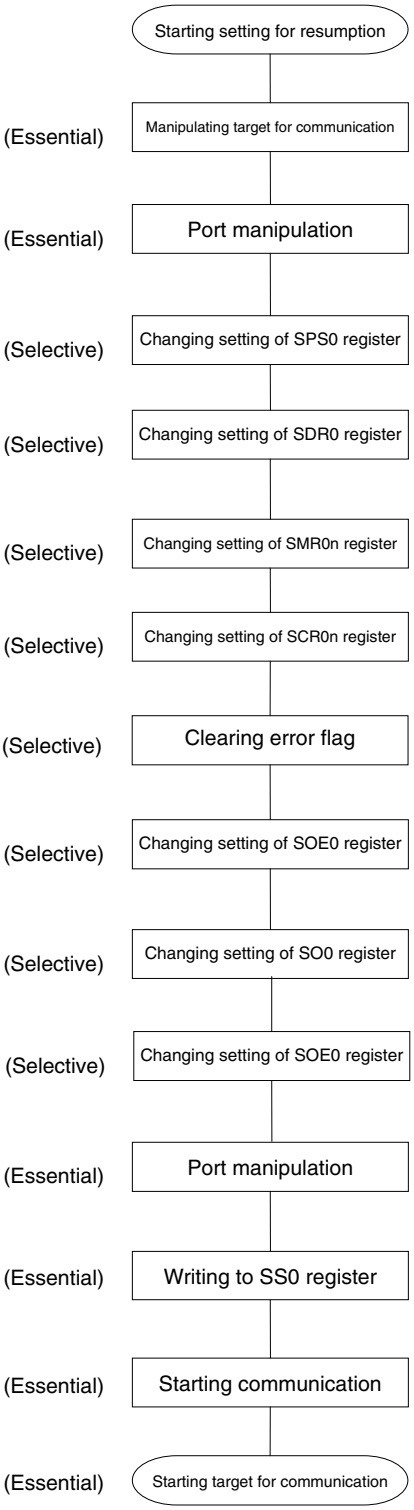
Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Figure 11-62. Procedure for Stopping Slave Transmission/Reception



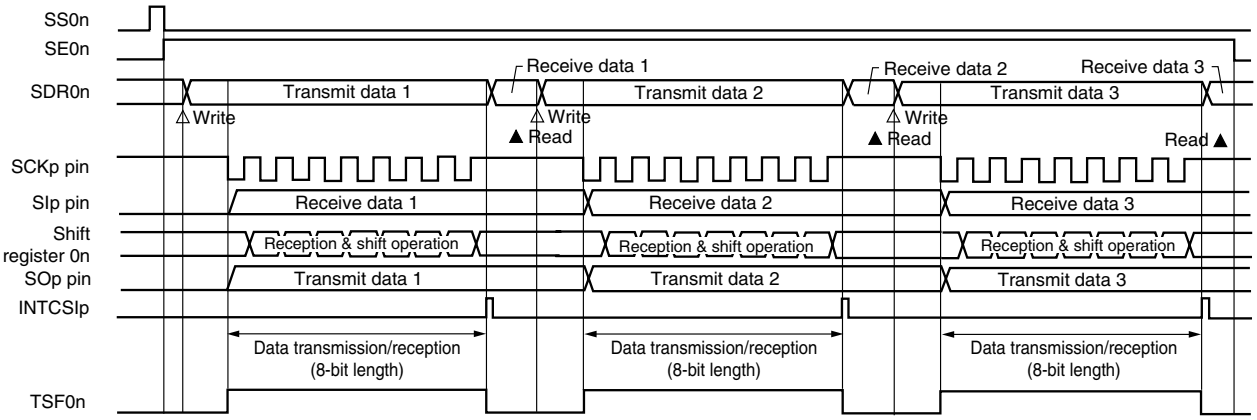
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SO0 register (see **Figure 11-63 Procedure for Resuming Slave Transmission/Reception**).

Figure 11-63. Procedure for Resuming Slave Transmission/Reception



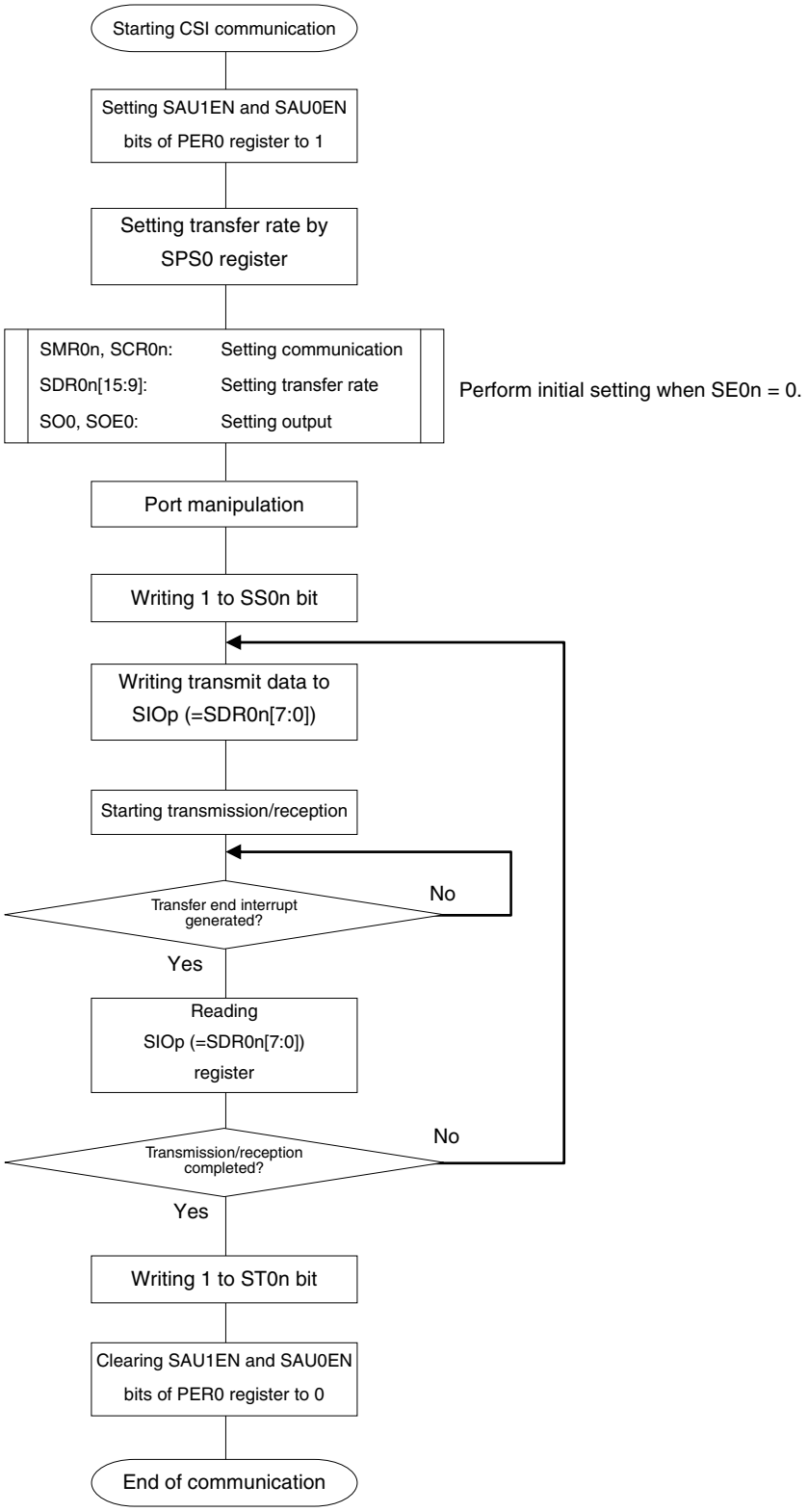
(3) Processing flow (in single-transmission/reception mode)

<R> **Figure 11-64. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)**



Remark n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)

Figure 11-65. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)

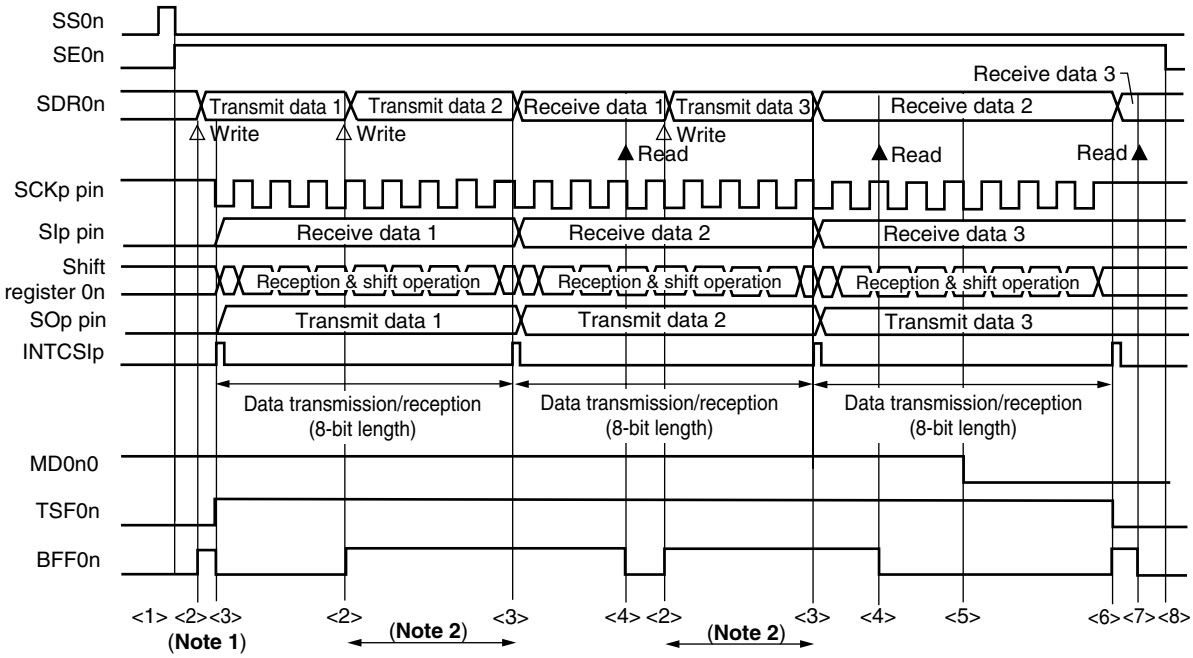


Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission/reception mode)

Figure 11-66. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)

<R>



Notes 1. When transmit data is written to the SDR0n register while BFF0n = 1, the transmit data is overwritten.

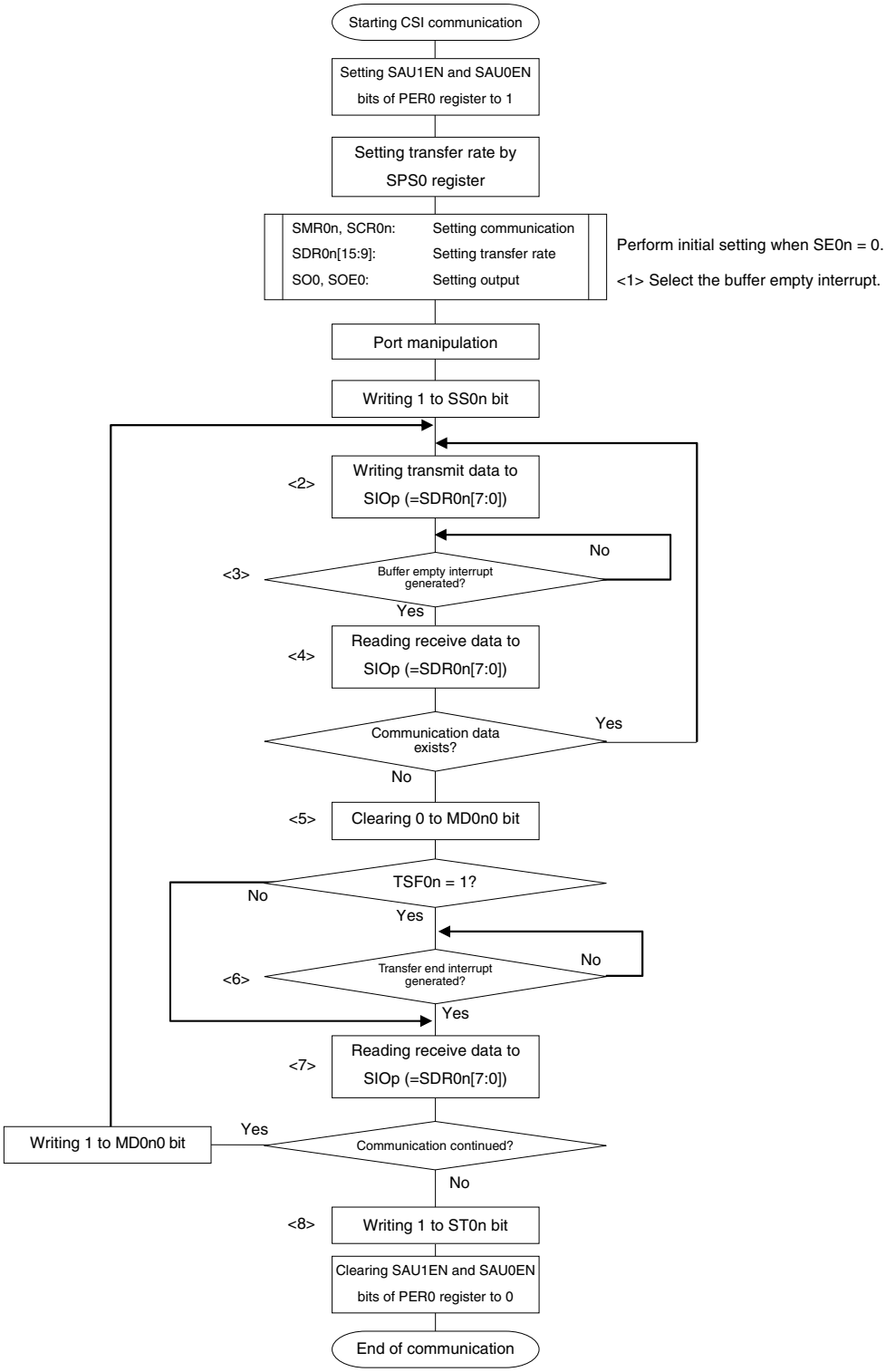
2. The transmit data can be read by reading the SDR0n register during this period. At this time, the transfer operation is not affected.

Caution The MD0n0 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 11-67 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. n: Channel number (n = 0, 2), p: CSI number (p = 00, 10)

Figure 11-67. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



<R>

Cautions After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 11-66 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, CSI10) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (MCK) frequency of target channel} ÷ (SDR0n[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master}[Note] [Hz]

Note The permissible maximum frequency is the smaller of $f_{CLK}/6$ [MHz] and $f_{MCK}/2$ [MHz].

Remarks 1. The value of SDR0n[15:9] is the value of bits 15 to 9 of the SDR0n register (0000000B to 1111111B) and therefore is 0 to 127.

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

The operation clock (MCK) is determined by serial clock select register 0 (SPS0) and bit 15 (CKS0n) of serial mode register 0n (SMR0n).

Table 11-2 Operating Clock Selection

SMR0n Register	SPS0 Register								Operation Clock (MCK) Note 1	
CKS0n	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000		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 Note 2	
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 Note 2	
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 (ST0 = 000FH) the operation of the serial array unit (SAU). When selecting INTTM02 for the operation clock, also stop the timer array unit (TAU) (TT0 = 00FFH).

2. SAU can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by setting the TIS02 bit of the TIS0 register of TAU to 1, selecting $f_{SUB}/4$ for the input clock, and selecting INTTM02 using the SPS0 register. When changing f_{CLK}, however, SAU and TAU must be stopped as described in Note 1 above.

Remarks 1. X: Don't care

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

11.6 Operation of UART (UART0, UART1, UART3) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. It transmits or receives data in asynchronization with the party of communication (by using an internal baud rate). Full-duplex UART communication can be realized by using two channels, one dedicated to transmission (even channel) and the other to reception (odd channel).

[Data transmission/reception]

- Data length of 5, 7, or 8 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

The LIN-bus is supported in UART3 (2, 3 channels of unit 1)

[LIN-bus functions]

- Wakeup signal detection
- Sync break field (SBF) detection
- Sync field measurement, baud rate calculation

} External interrupt (INTP0) or timer array unit (TAU) is used.

UART0 uses channels 0 and 1 of SAU0.
UART1 uses channels 2 and 3 of SAU1.
UART3 uses channels 2 and 3 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	–
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	–	–
	1	–	–	–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

UART performs the following four types of communication operations.

- UART transmission (See **11.6.1**.)
- UART reception (See **11.6.2**.)
- LIN transmission (UART3 only) (See **11.6.3**.)
- LIN reception (UART3 only) (See **11.6.4**.)

11.6.1 UART transmission

UART transmission is an operation to transmit data from the 78K0R/KE3 to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART0	UART1	UART3
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 2 of SAU1
Pins used	TxD0	TxD1	TxD3
Interrupt	INTST0	INTST1	INTST3
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	None		
Transfer data length	5, 7, or 8 bits		
Transfer rate	Max. $f_{MCK}/6$ [bps] (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 27 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-68. Example of Contents of Registers for UART Transmission of UART (UART0, UART1, UART3) (1/2)

(a) Serial output register m (SOm) … Sets only the bit of the target channel to 1.

SO0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	CKO02 ×	1	CKO00 ×	0	0	0	0	1	SO02 0/1 Note	1	SO00 0/1 Note

SO1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1	0	0	0	0	1	SO12 0/1 Note	1	1

(b) Serial output enable register m (SOEm) … Sets only the bit of the target channel to 1.

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	0	SOE00 0/1

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	SOE12 0/1	0	0

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

SS0

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	SS03 ×	SS02 0/1	SS01 ×	SS00 0/1

SS1

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	SS13 ×	SS12 0/1	0	0

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), mn = 00, 02, 12

□ : 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 11-68. Example of Contents of Registers for UART Transmission of UART (UART0, UART1, UART3) (2/2)

(d) Serial output level register m (SOLm) … Sets only the bits of the target channel.

SOL0

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	SOL02 0/1	0	SOL00 0/1

0: Forward (normal) transmission
1: Reverse transmission

SOL1

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	SOL12 0/1	0	0

0: Forward (normal) transmission
1: Reverse transmission

(e) Serial mode register mn (SMRmn)

SMRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0/1

Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(f) Serial communication operation setting register mn (SCRmn)

SCRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0/1	PTCmn0 0/1	DIRmn 0/1	0	SLCmn1 0/1	SLCmn0 0/1	0	DLSmn2 1	DLSmn1 0/1	DLSmn0 0/1

Setting of parity bit
00B: No parity
01B: 0 parity
10B: Even parity
11B: Odd parity

Setting of stop bit
01B: Appending 1 bit
10B: Appending 2 bits

(g) Serial data register mn (SDRmn) (lower 8 bits: TXDq)

SDRmn

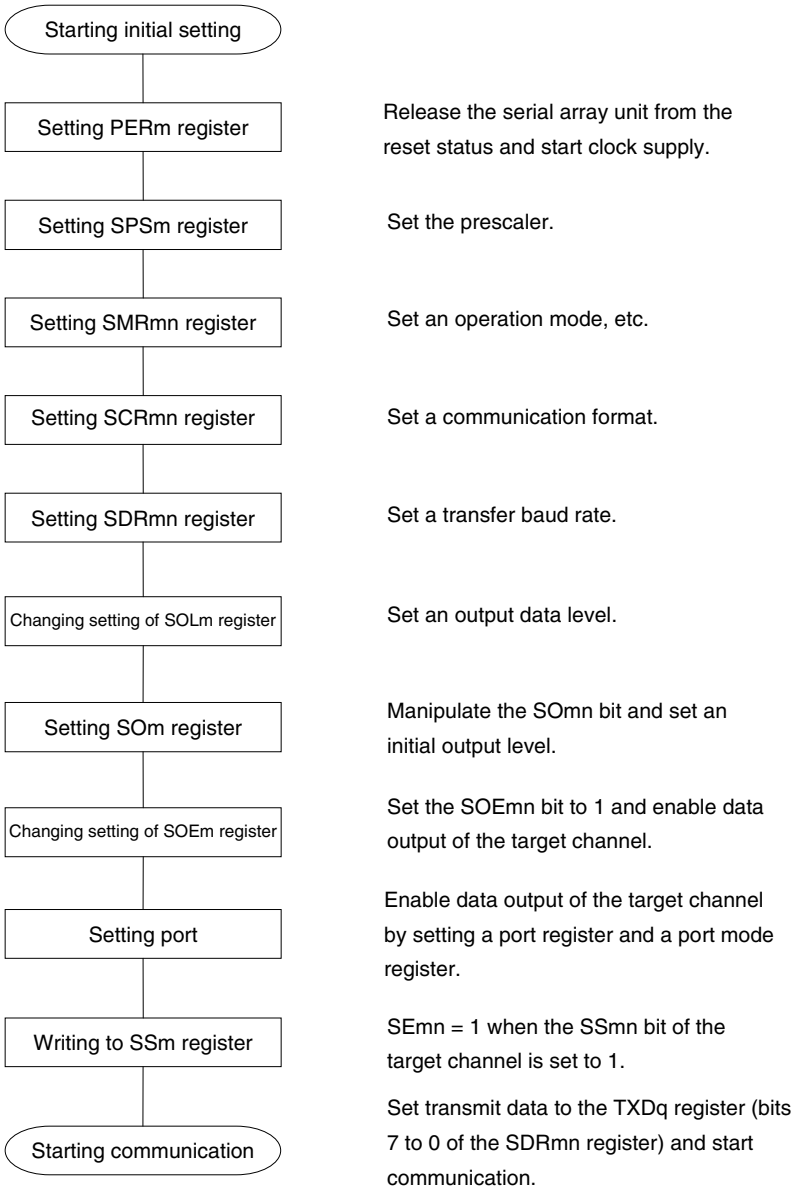
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Baud rate setting							0	Transmit data setting							

TXDq

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12,
q: UART number (q = 0, 1, 3)
□: Setting is fixed in the UART transmission mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

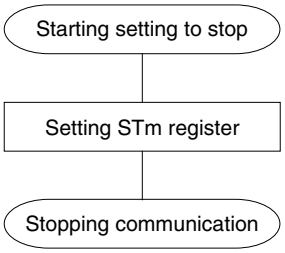
(2) Operation procedure

Figure 11-69. Initial Setting Procedure for UART Transmission



Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-70. Procedure for Stopping UART Transmission

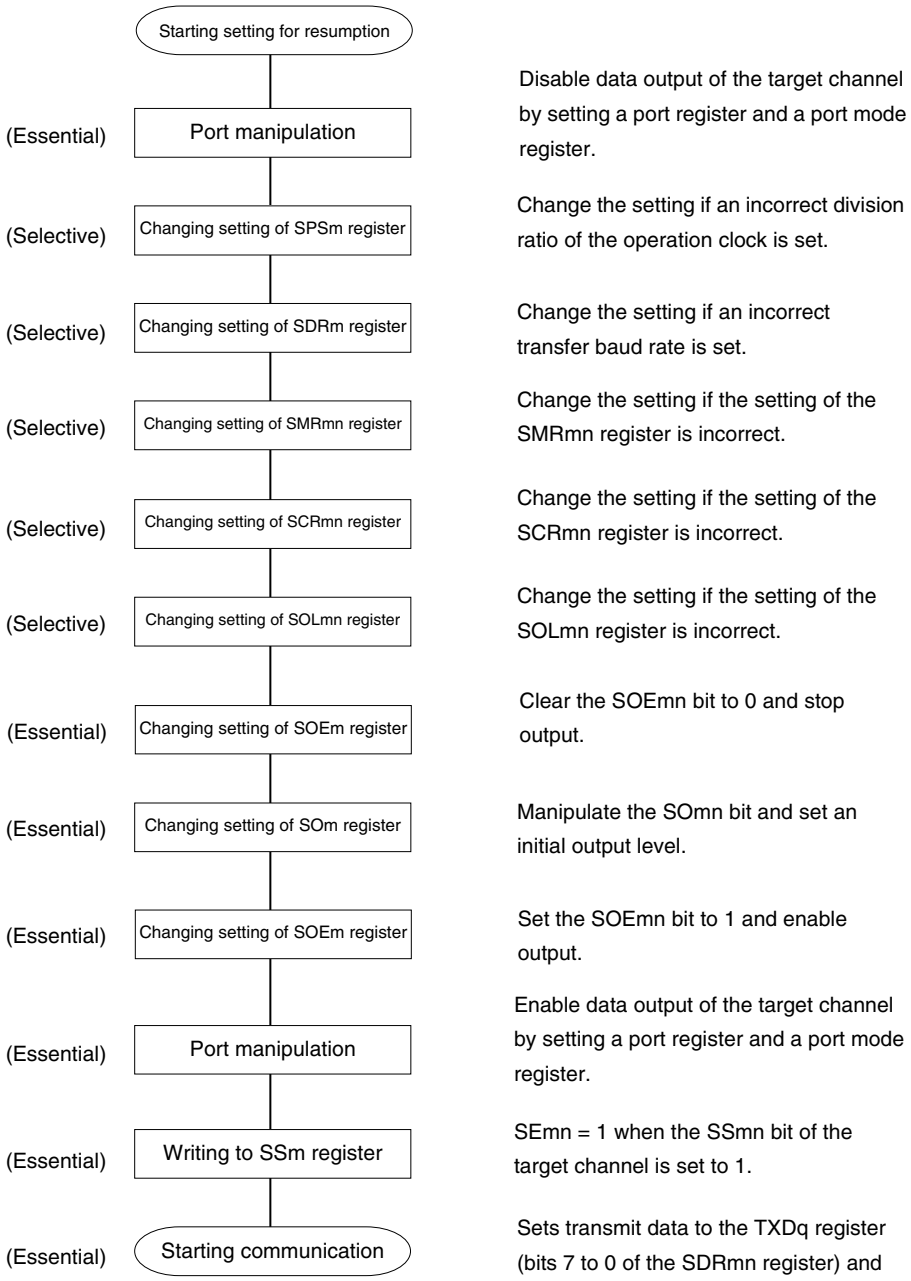


Write 1 to the STmn bit of the target channel.

Stop communication in midway.

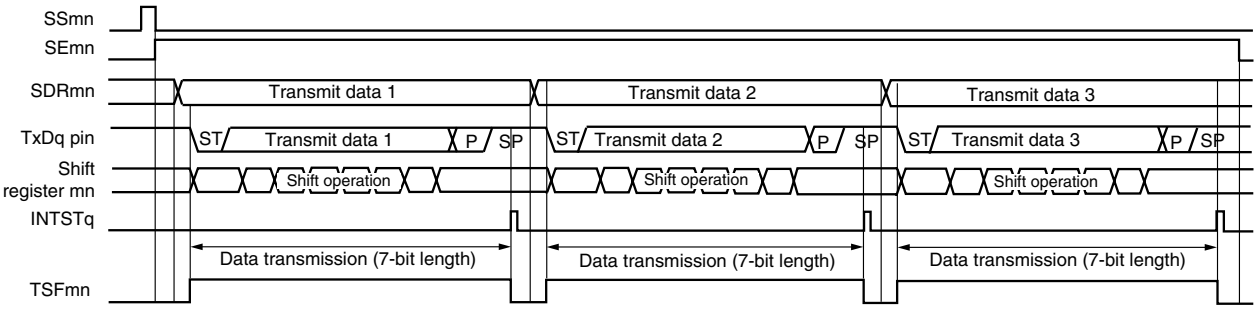
Remark Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 11-71 Procedure for Resuming UART Transmission**).

Figure 11-71. Procedure for Resuming UART Transmission



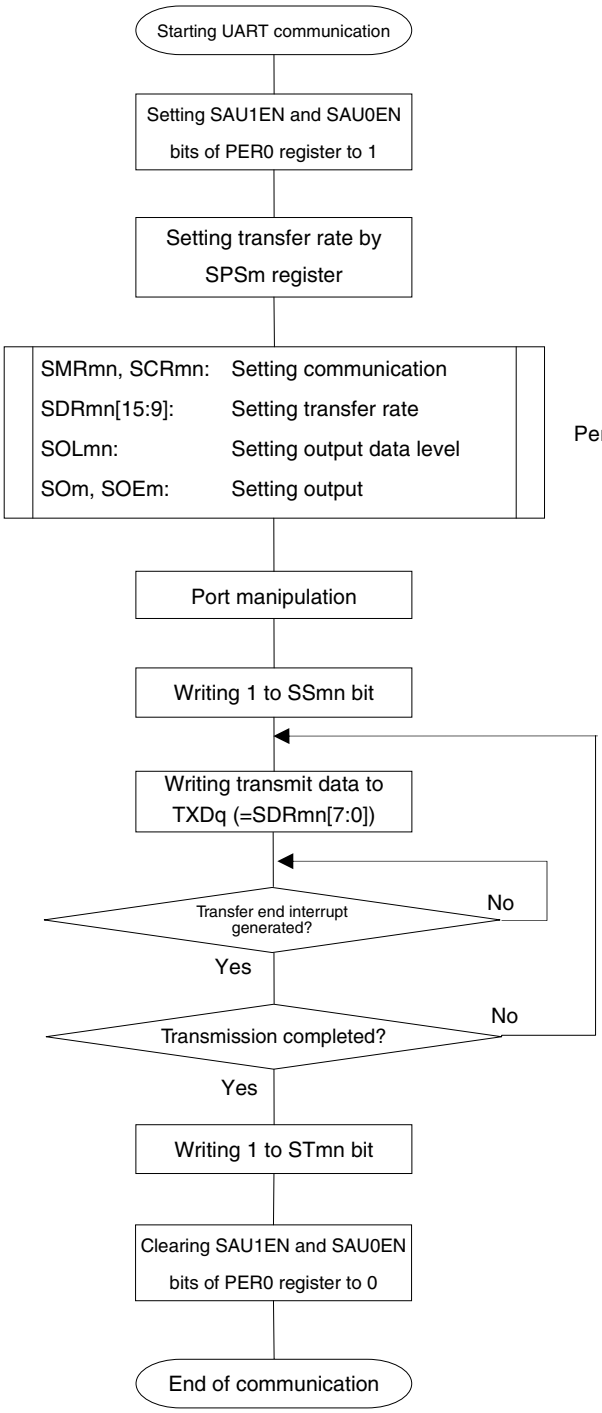
(3) Processing flow (in single-transmission mode)

Figure 11-72. 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, 12,
q: UART number (q = 0, 1, 3)

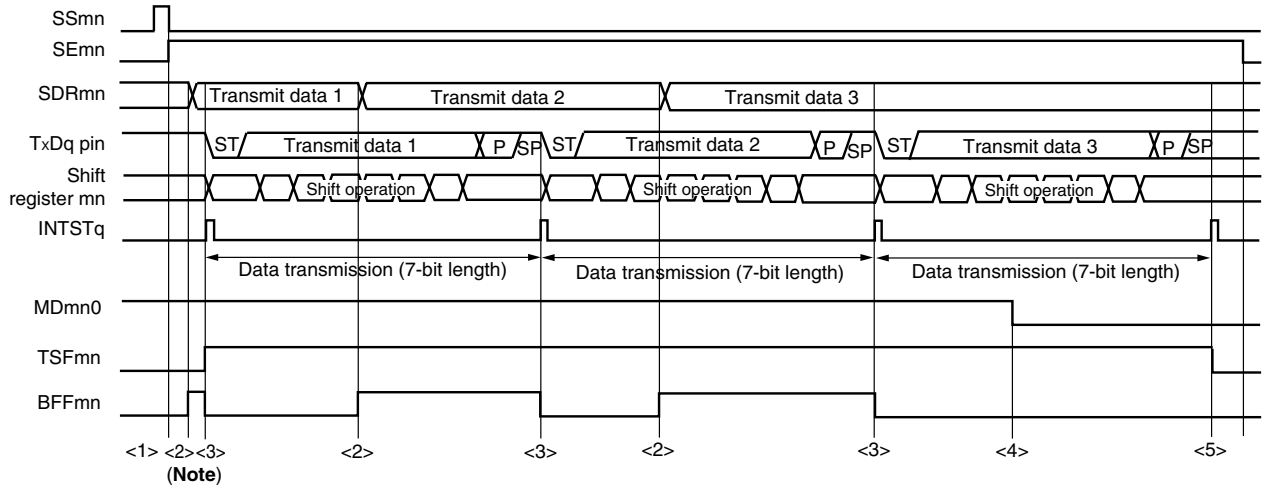
Figure 11-73. Flowchart of UART Transmission (in Single-Transmission Mode)



Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

(4) Processing flow (in continuous transmission mode)

Figure 11-74. Timing Chart of UART Transmission (in Continuous Transmission Mode)

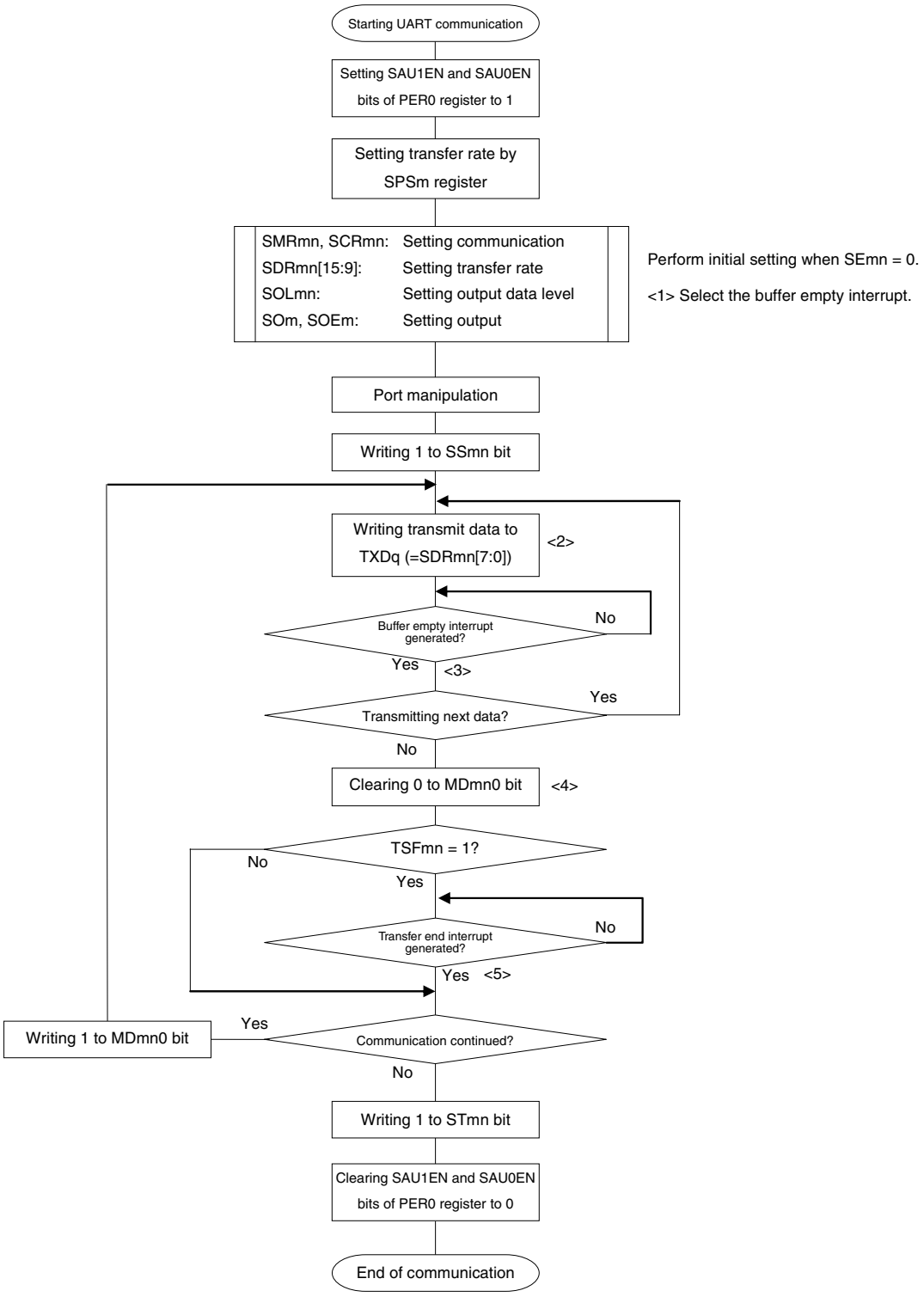


Note When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

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, 12,
q: UART number (q = 0, 1, 3)

Figure 11-75. Flowchart of UART Transmission (in Continuous Transmission Mode)



Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark <1> to <5> in the figure correspond to <1> to <5> in **Figure 11-74 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

11.6.2 UART reception

UART reception is an operation wherein the 78K0R/KE3 asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd channel of the two channels used for UART is used.

UART	UART0	UART1	UART3
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 3 of SAU1
Pins used	RxD0	RxD1	RxD3
Interrupt	INTSR0	INTSR1	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error interrupt	INTSRE0	INTSRE1	INTSRE3
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)		
<R> 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 27 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

(1) Register setting

Figure 11-76. Example of Contents of Registers for UART Reception of UART (UART0, UART1, UART3) (1/2)

(a) Serial output register m (SOm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 ×	1	CKO00 ×	0	0	0	0	1	SO02 ×	1	SO00 ×
SO1	0	0	0	0	1	1	1	1	0	0	0	0	1	SO12 ×	1	1

(b) Serial output enable register m (SOEm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 ×	0	SOE00 ×
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12 ×	0	0

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 0/1	SS02 ×	SS01 0/1	SS00 ×
SS1	0	0	0	0	0	0	0	0	0	0	0	0	SS13 0/1	SS12 ×	0	0

(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 1	0	SISmn0 0/1	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0

0: Forward (normal) reception
1: Reverse reception

Operation mode of channel n
0: Transfer end interrupt

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, 13
r: Channel number (r = n – 1),
□ : 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 11-76. Example of Contents of Registers for UART Reception of UART (UART0, UART1, UART3) (2/2)

(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

Operation mode of channel r
0: Transfer end interrupt
1: Buffer empty interrupt

(f) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0		DLSmn2	DLSmn1	DLSmn0
	0	1	0	0	0	1	0/1	0/1	0/1	0	0	1	0	1	0/1	0/1

(g) Serial data register mn (SDRmn) (lower 8 bits: RXDq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Receive data register							

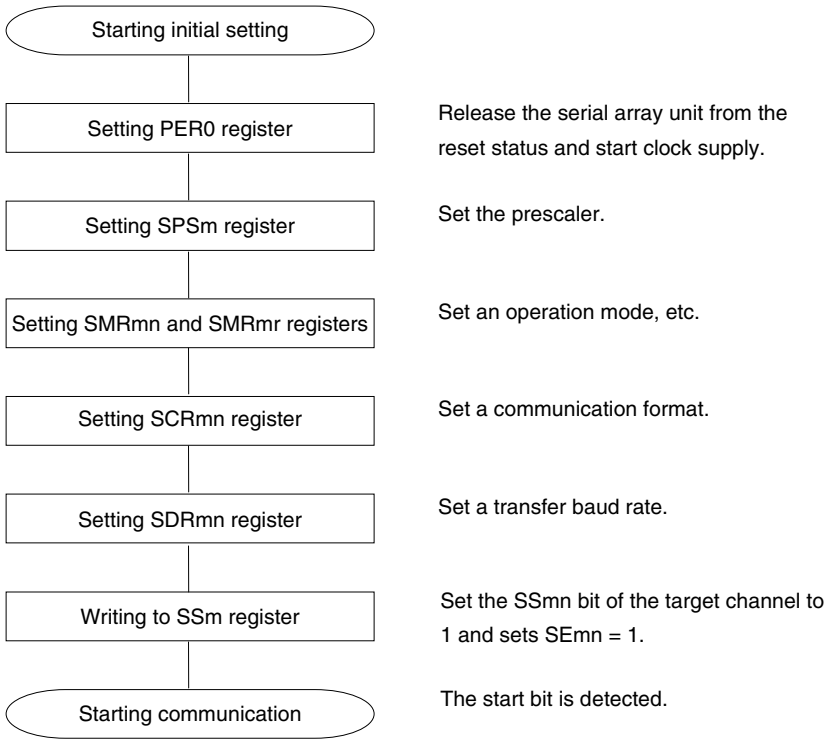
RXDq

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, 13,
r: Channel number (r = n – 1), q: UART number (q = 0, 1, 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 11-77. Initial Setting Procedure for UART Reception



Cautions After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-78. Procedure for Stopping UART Reception

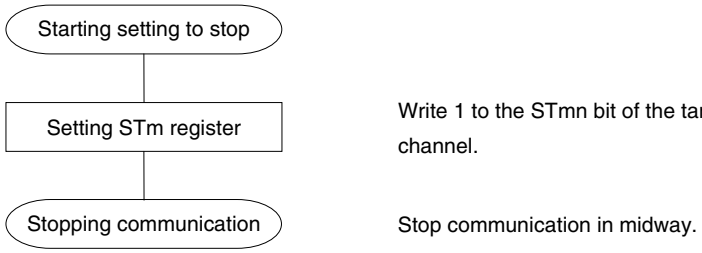
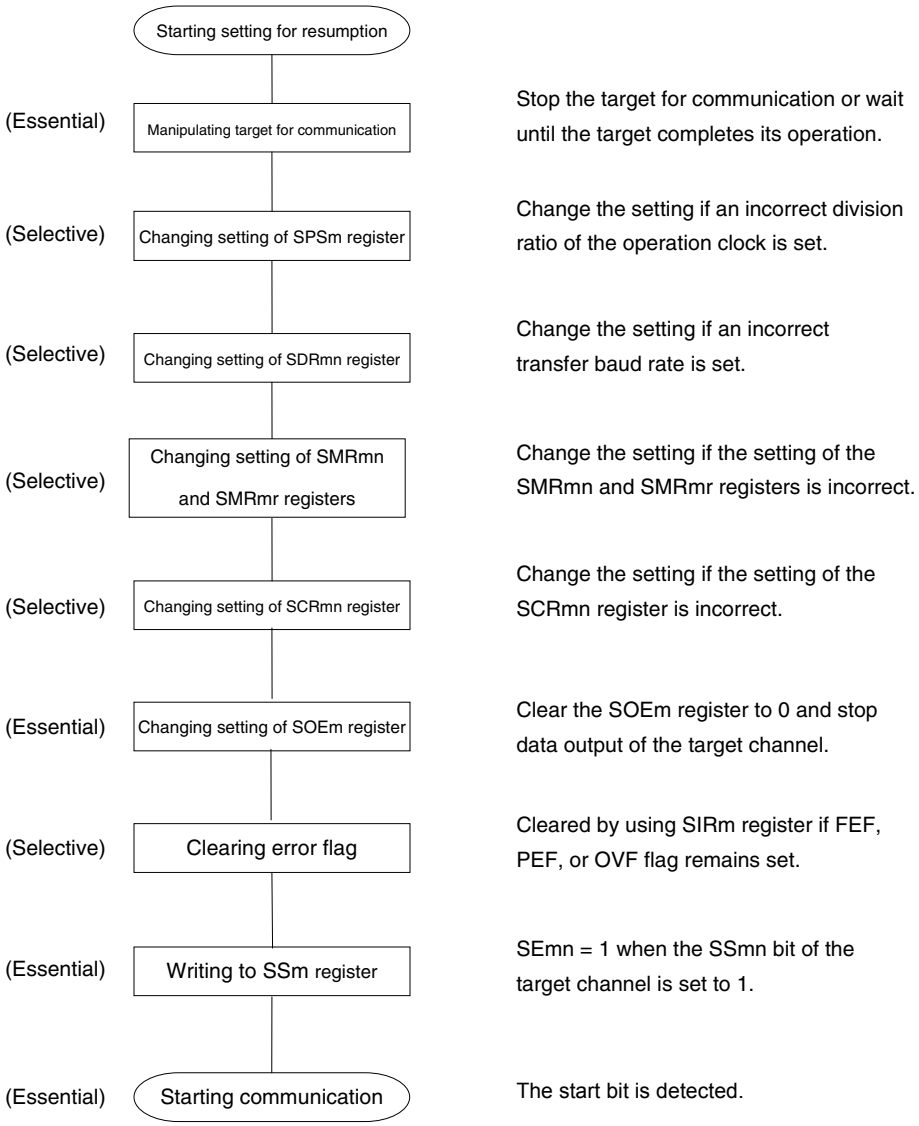


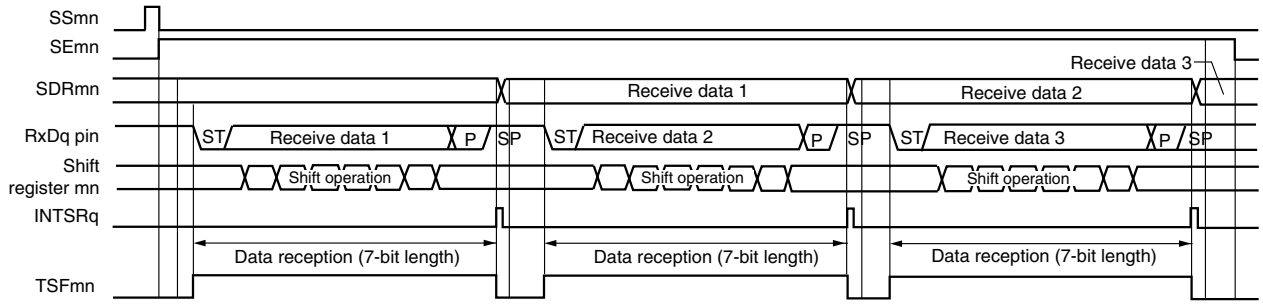
Figure 11-79. Procedure for Resuming UART Reception



(3) Processing flow

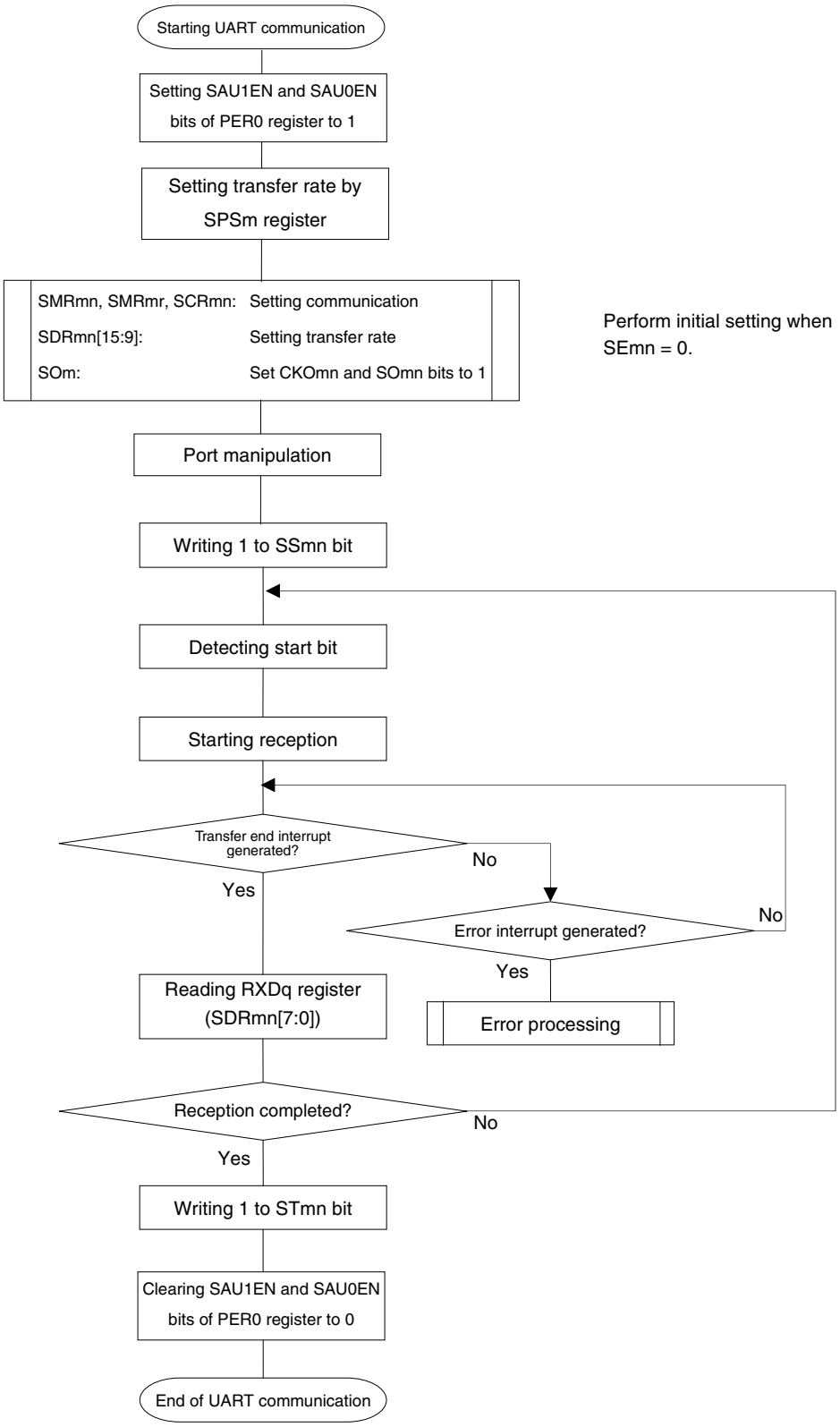
Figure 11-80. Timing Chart of UART Reception

<R>



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 13,
q: UART number (q = 0, 1, 3)

Figure 11-81. Flowchart of UART Reception



Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

11.6.3 LIN transmission

Of UART transmission, UART3 supports LIN communication.

For LIN transmission, channel 2 of unit 1 (SAU1) is used.

UART	UART0	UART1	UART3
Support of LIN communication	Not supported	Not supported	Supported
Target channel	–	–	Channel 2 of SAU1
Pins used	–	–	TxD3
Interrupt	–	–	INTST3
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	None		
<R> Transfer data length	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 27 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network.

Communication of LIN is single-master communication and up to 15 slaves can be connected to one master.

The slaves are used to control switches, actuators, and sensors, which are connected to the master via LIN.

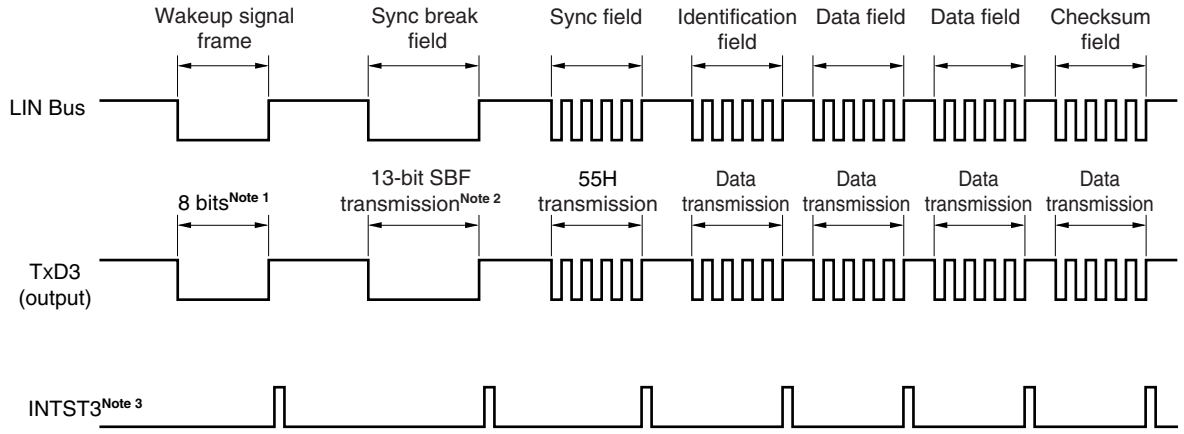
Usually, the master is connected to a network such as CAN (Controller Area Network).

A LIN bus is a single-wire bus to which nodes are connected via transceiver conforming to ISO9141.

According to the protocol of LIN, the master transmits a frame by attaching baud rate information to it. A slave receives this frame and corrects a baud rate error from the master. If the baud rate error of a slave is within ±15%, communication can be established.

Figure 11-82 outlines a transmission operation of LIN.

Figure 11-82. Transmission Operation of LIN



Notes 1. The baud rate is set so as to satisfy the standard of the wakeup signal and data of 00H is transmitted.

2. A sync break field is defined to have a width of 13 bits and output a low level. Where the baud rate for main transfer is N [bps], therefore, the baud rate of the sync break field is calculated as follows.

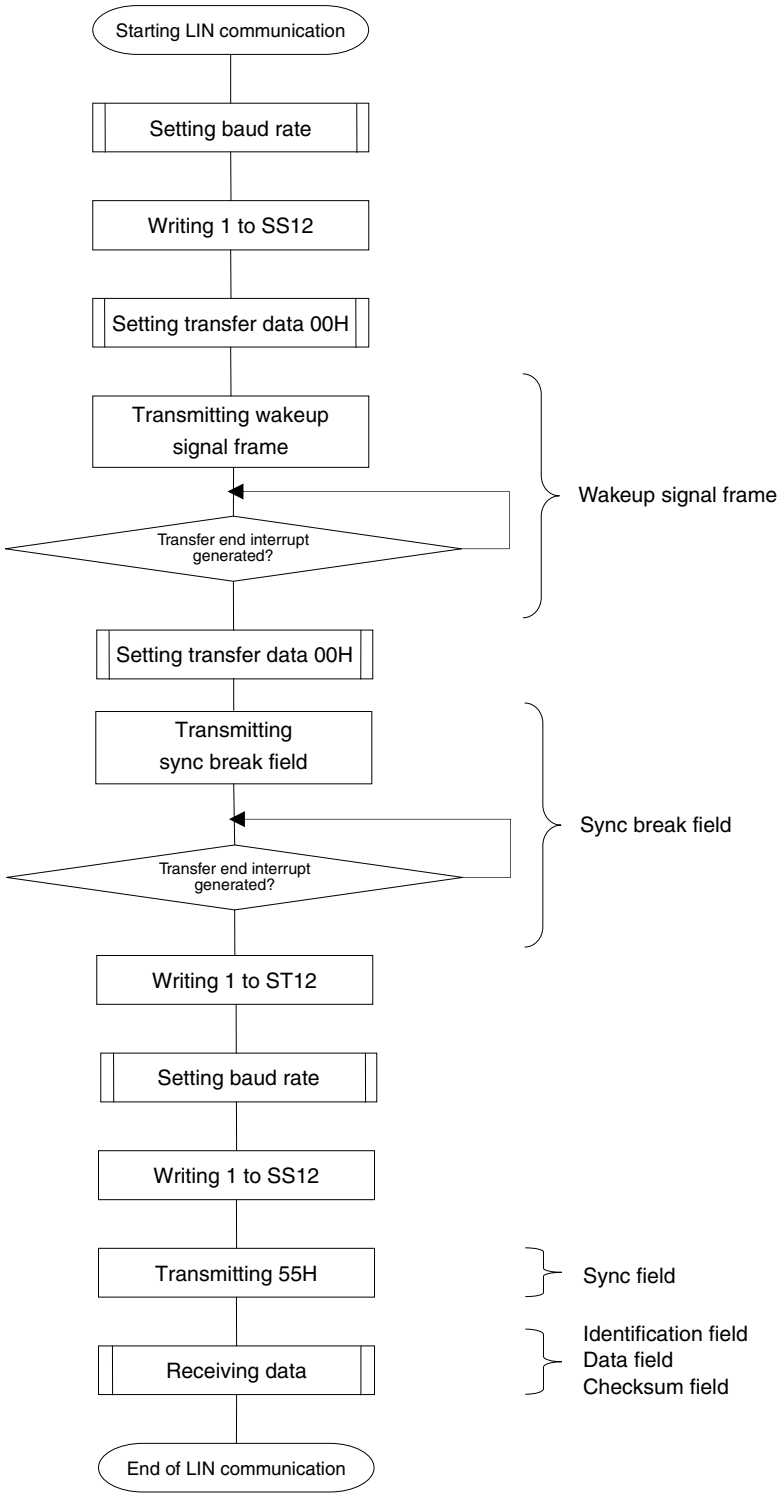
(Baud rate of sync break field) = 8/13 × N

By transmitting data of 00H at this baud rate, a sync break field is generated.

3. INTST3 is output upon completion of transmission. INTST3 is also output when SBF transmission is executed.

Remark The interval between fields is controlled by software.

Figure 11-83. Flowchart for LIN Transmission



11.6.4 LIN reception

Of UART reception, UART3 supports LIN communication.

For LIN reception, channel 3 of unit 1 (SAU1) is used.

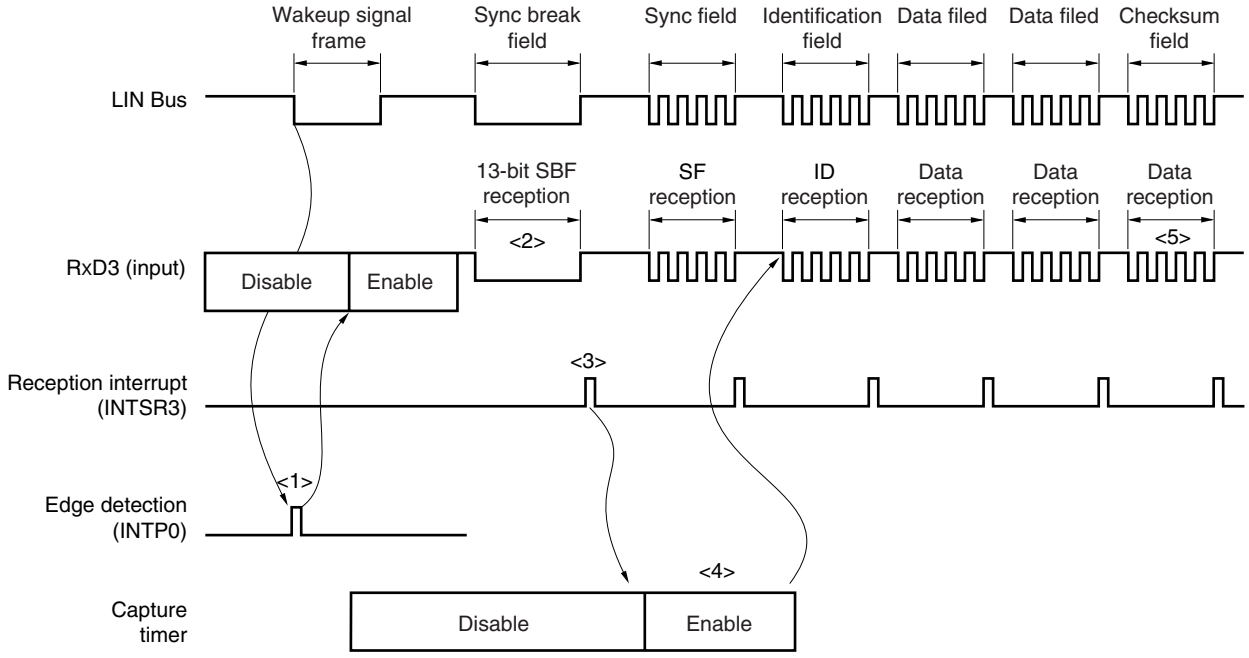
UART	UART0	UART1	UART3
Support of LIN communication	Not supported	Not supported	Supported
Target channel	–	–	Channel 0 of SAU1
Pins used	–	–	RxD3
Interrupt	–	–	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error interrupt	–	–	INTSRE3
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)		
<R> Transfer data length	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 27 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

Figure 11-84 outlines a reception operation of LIN.

Figure 11-84. Reception Operation of LIN



Here is the flow of signal processing.

<1> The wakeup signal is detected by detecting an interrupt edge (INTP0) on a pin. When the wakeup signal is detected, enable reception of UART3 (RXE13 = 1) and wait for SBF reception.

<2> When the start bit of SBF is detected, reception is started and serial data is sequentially stored in the RXD3 register (= bits 7 to 0 of the serial data register 13 (SDR13)) at the set baud rate. When the stop bit is detected, the reception end interrupt request (INTSR3) is generated. When data of low levels of 11 bits or more is detected as SBF, it is judged that SBF reception has been correctly completed. If data of low levels of less than 11 bits is detected as SBF, it is judged that an SBF reception error has occurred, and the system returns to the SBF reception wait status.

<3> When SBF reception has been correctly completed, start channel 7 of the timer array unit and measure the bit interval (pulse width) of the sync field (see **6.7.5 Operation as input signal high-/low-level width measurement**).

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART3 once and adjust (re-set) the baud rate.

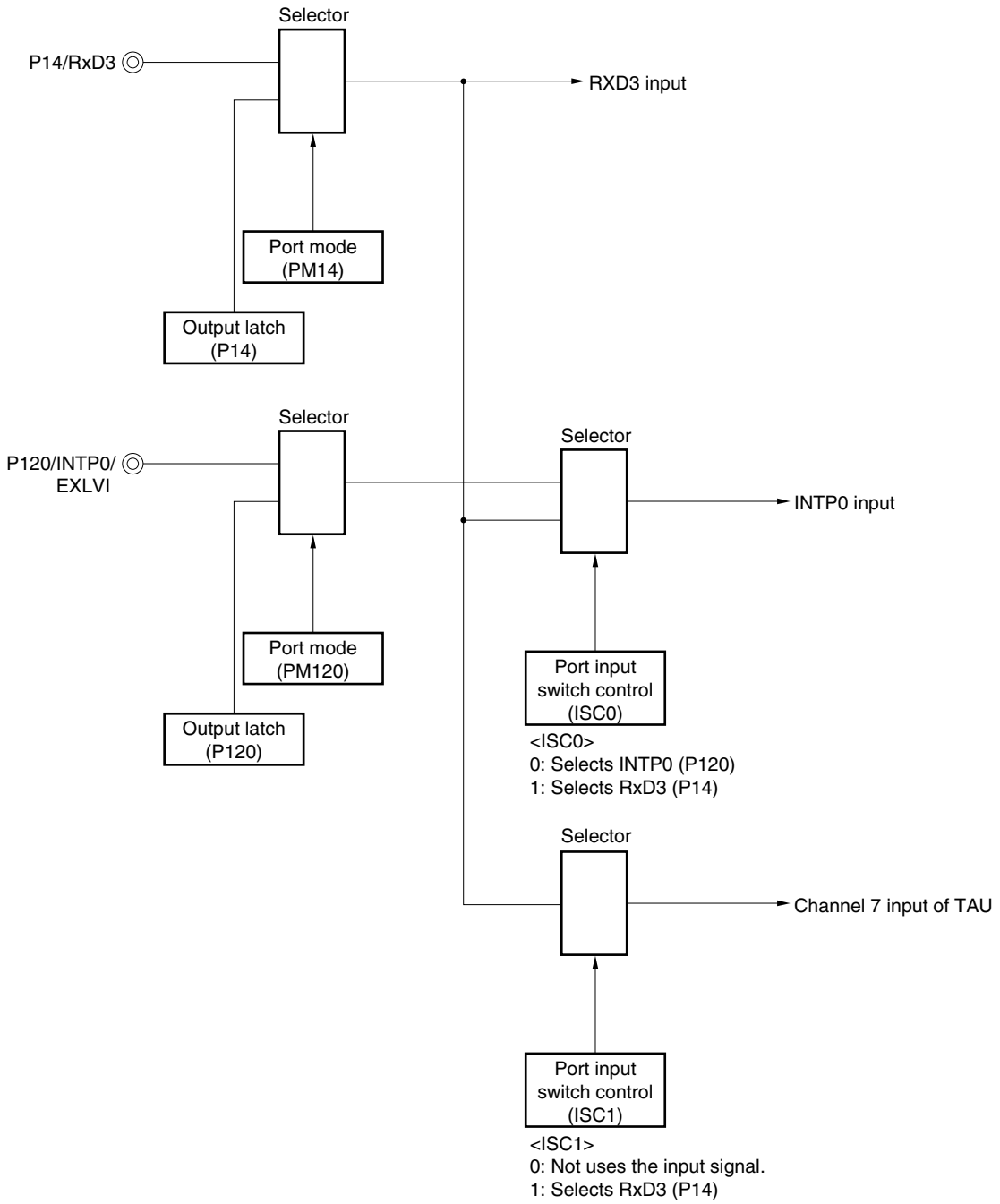
<5> The checksum field should be distinguished by software. In addition, processing to initialize UART3 after the checksum field is received and to wait for reception of SBF should also be performed by software.

Figure 11-85 shows the configuration of a port that manipulates reception of LIN.

The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (INTP0). The length of the sync field transmitted from the master can be measured by using the external event capture operation of the timer array unit (TAU) to calculate a baud-rate error.

By controlling switch of port input (ISC0/ISC1), the input source of port input (RxD3) for reception can be input to the external interrupt pin (INTP0) and timer array unit (TAU).

Figure 11-85. Port Configuration for Manipulating Reception of LIN



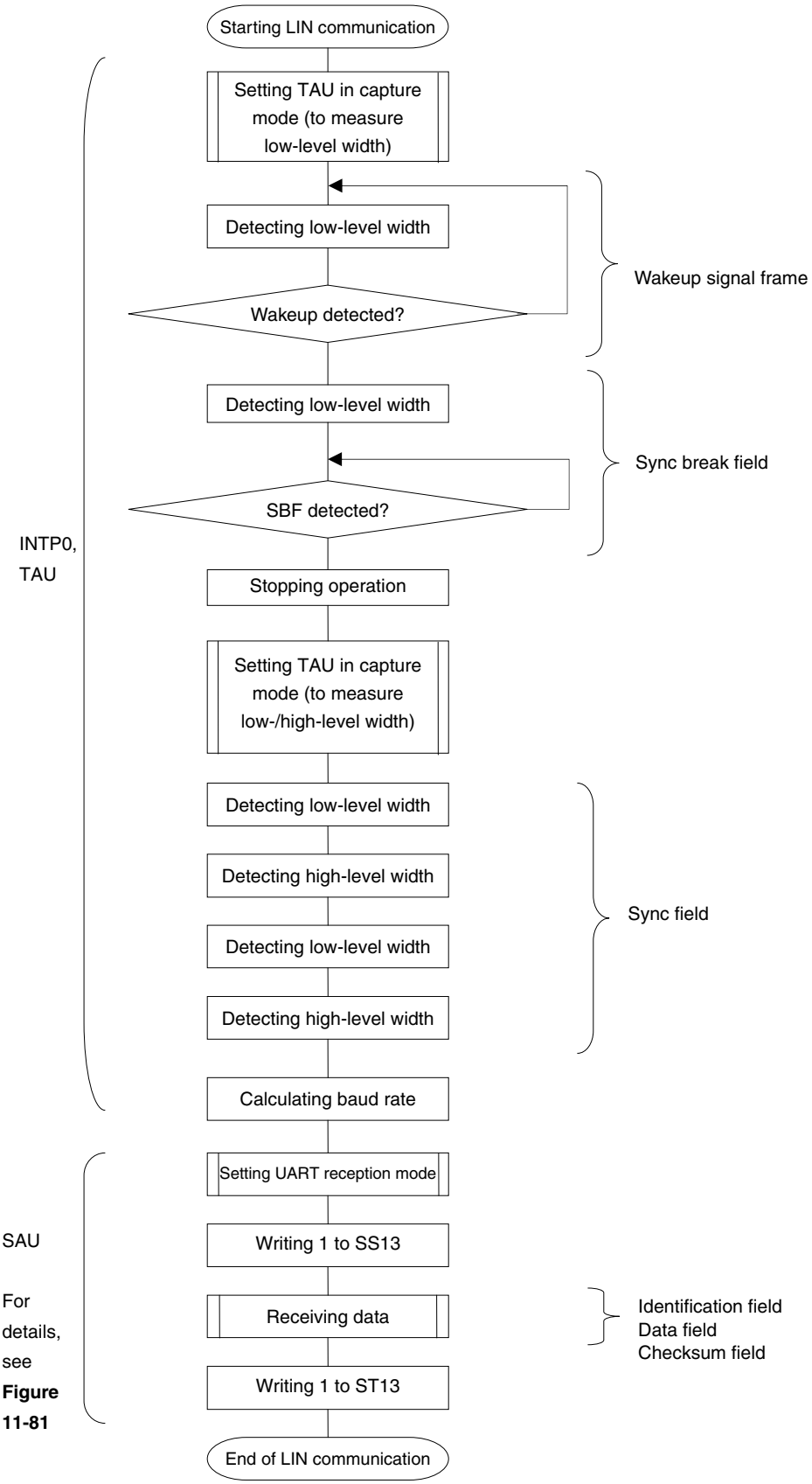
Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 11-17**.)

The peripheral functions used for the LIN communication operation are as follows.

<Peripheral functions used>

- External interrupt (INTP0); Wakeup signal detection
  Usage: To detect an edge of the wakeup signal and the start of communication
- Channel 7 of timer array unit (TAU); Baud rate error detection
  Usage: To detect the length of the sync field (SF) and divide it by the number of bits in order to detect an error (The interval of the edge input to RxD3 is measured in the capture mode.)
- Channels 2 and 3 (UART3) of serial array unit 1 (SAU1)

Figure 11-86. Flowchart of LIN Reception



11.6.5 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0, UART1, UART3) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (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 = 00-03, 12, 13

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 11-3 Operating Clock Selection

SMRmn Register	SPSm Register								Operation Clock (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	If m = 0: INTTM02, if m = 1: INTTM03 [Note 2]	
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	If m = 0: INTTM02, if m = 1: INTTM03 [Note 2]	
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 and INTTM03 for the operation clock, also stop the timer array unit (TAU) (TT0 = 00FFH).

2. SAU can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by setting the TIS02 (if m = 0) and TIS03 (if m = 1) bits of the TIS0 register of TAU to 1, selecting $f_{SUB}/4$ for the input clock, and selecting INTTM02 and INTTM03 using the SPSm register. When changing f_{CLK}, however, SAU and TAU 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, 12, 13

(2) Baud rate error during transmission

The baud rate error of UART (UART0, UART1, UART3) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 − 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 20 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 20 MHz			
	Operation Clock (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 %

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0, UART1, UART3) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

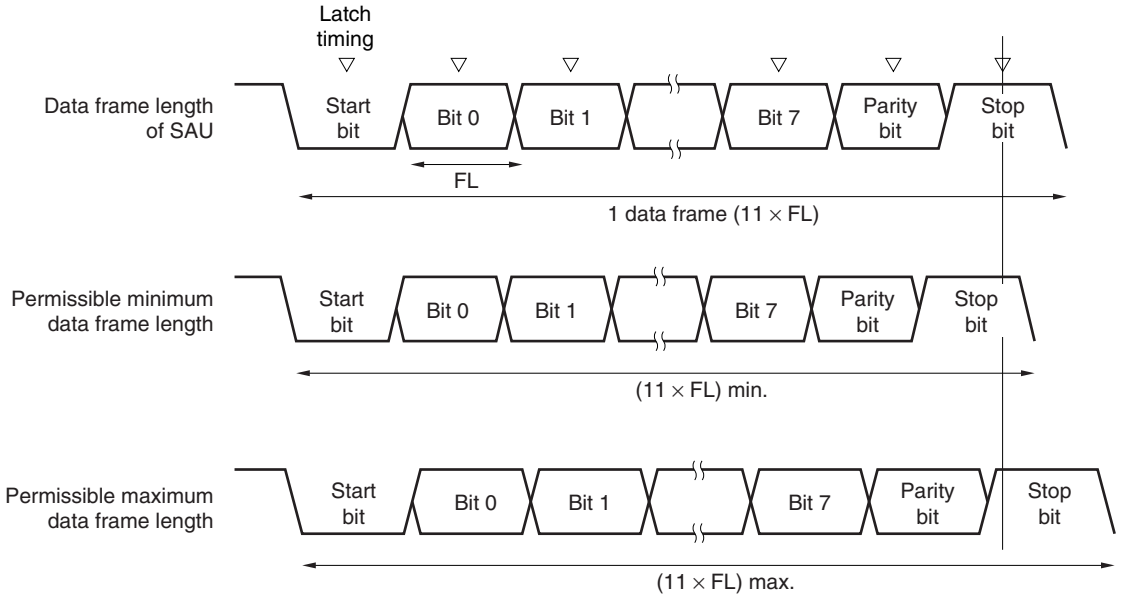
Brate: Calculated baud rate value at the reception side (See **11.6.5 (1) Baud rate calculation expression**.)

k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3)

Figure 11-87. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 11-87, 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.

11.7 Operation of Simplified I²C (IIC10) 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 and ACK detection functions
- 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 function

Remark To use an I²C bus of full function, see **CHAPTER 12 SERIAL INTERFACE IIC0**.

The channels supporting simplified I²C (IIC10) are channel 2 of SAU0.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	–
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	–	–	–
	1	–	–	–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

Simplified I²C (IIC10) 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
Target channel	Channel 2 of SAU0
Pins used	SCL10, SDA10
Interrupt	INTIIC10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Parity error detection flag (PEF02)
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_{CLK}/4 MHz f_{CLK}: System clock frequency 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

(1) Register setting

Figure 11-88. Example of Contents of Registers for Address Field Transmission of Simplified I^2C (IIC10)

(a) Serial output register 0 (SO0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 0/1	1	CKO00 ×	0	0	0	0	1	SO02 0/1	1	SO00 ×

Start condition is generated by manipulating the SO02 bit.

(b) Serial output enable register 0 (SOE0) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02 0/1	0	SOE00 ×

SOE02 = 0 until the start condition is generated, and SOE02 = 1 after generation.

(c) Serial channel start register 0 (SS0) … Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 ×

(d) Serial mode register 02 (SMR02)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMR02	CKS02 0/1	CCS02 0	0	0	0	0	0	STS02 0	0	SIS020 0	1	0	0	MD022 1	MD021 0	MD020 0

Operation mode of channel 2
0: Transfer end interrupt

(e) Serial communication operation setting register 02 (SCR02)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR02	TXE02 1	RXE02 0	DAP02 0	CKP02 0	0	EOC02 0	PTC021 0	PTC020 0	DIR02 0	0	SLC021 0	SLC020 1	0	DLS022 1	DLS021 1	DLS020 1

Setting of parity bit
00B: No parity

Setting of stop bit
01B: Appending 1 bit (ACK)

(f) Serial data register 02 (SDR02) (lower 8 bits: SIO10)

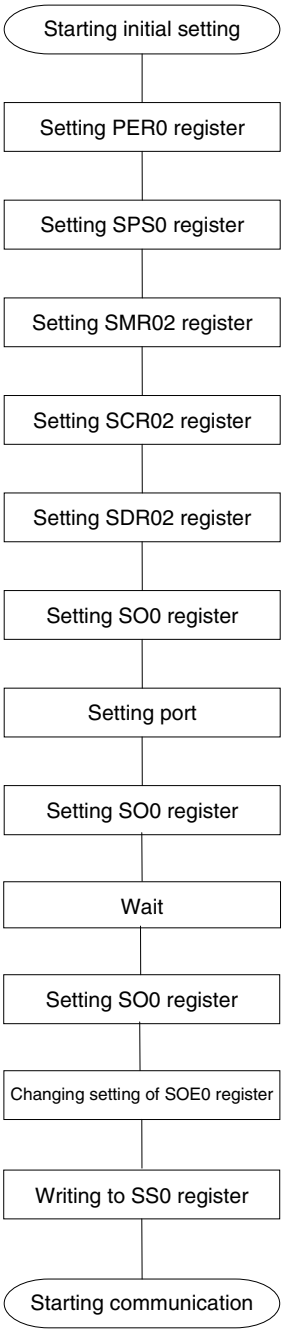
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR02	Baud rate setting							0	Transmit data setting (address + R/W)							

SIO10

Remark □: 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 11-89. Initial Setting Procedure for Address Field Transmission



Release the serial array unit from the reset status and start clock supply.

Set the prescaler.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate.

<R> Manipulate the SO02 and CKO02 bit and set an initial output level.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Clear the SO02 bit to 0 to generate the start condition.

<R> Secure a wait time so that the specifications of I^2C on the slave side are satisfied.

<R> Clear the CKO02 bit to 0 to lower the clock output level.

<R> Set the SOE02 bit to 1 and enable data output of the target channel.

SE02 = 1 when the SS02 bit of the target channel is set to 1.

Set address and R/W to the SIO10 register (bits 7 to 0 of the SDR02 register) and start communication.

Caution After setting the PER0 register to 1, be sure to set the SPS0 register after 4 or more clocks have elapsed.

(3) Processing flow

Figure 11-90. Timing Chart of Address Field Transmission

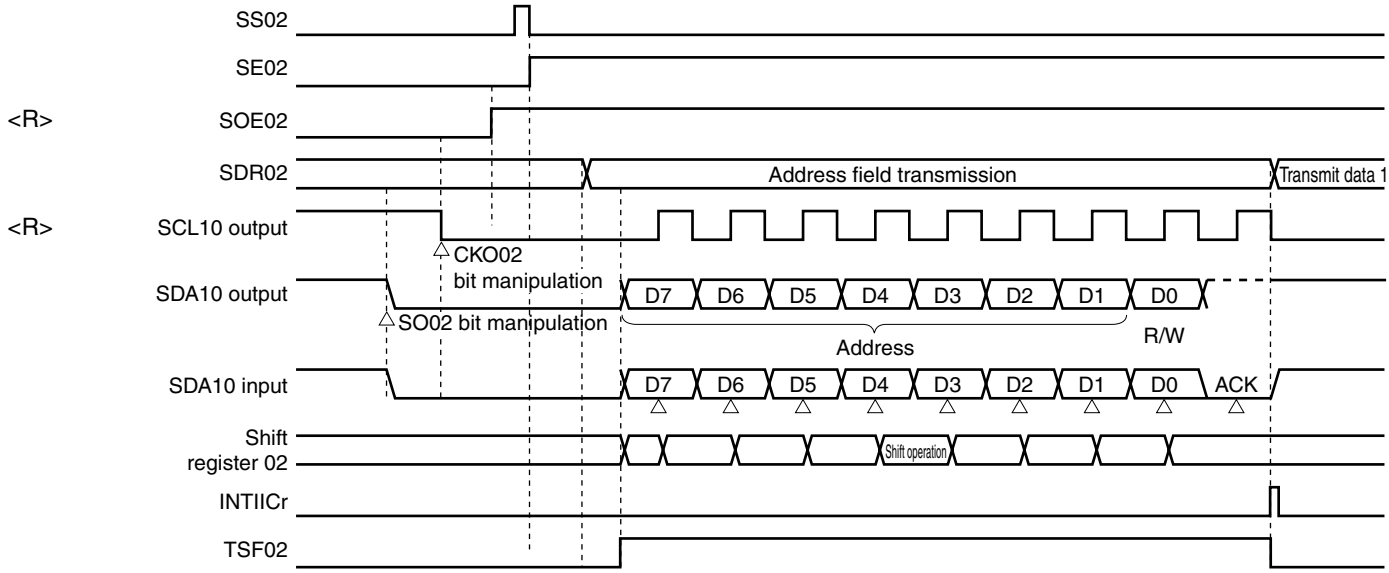
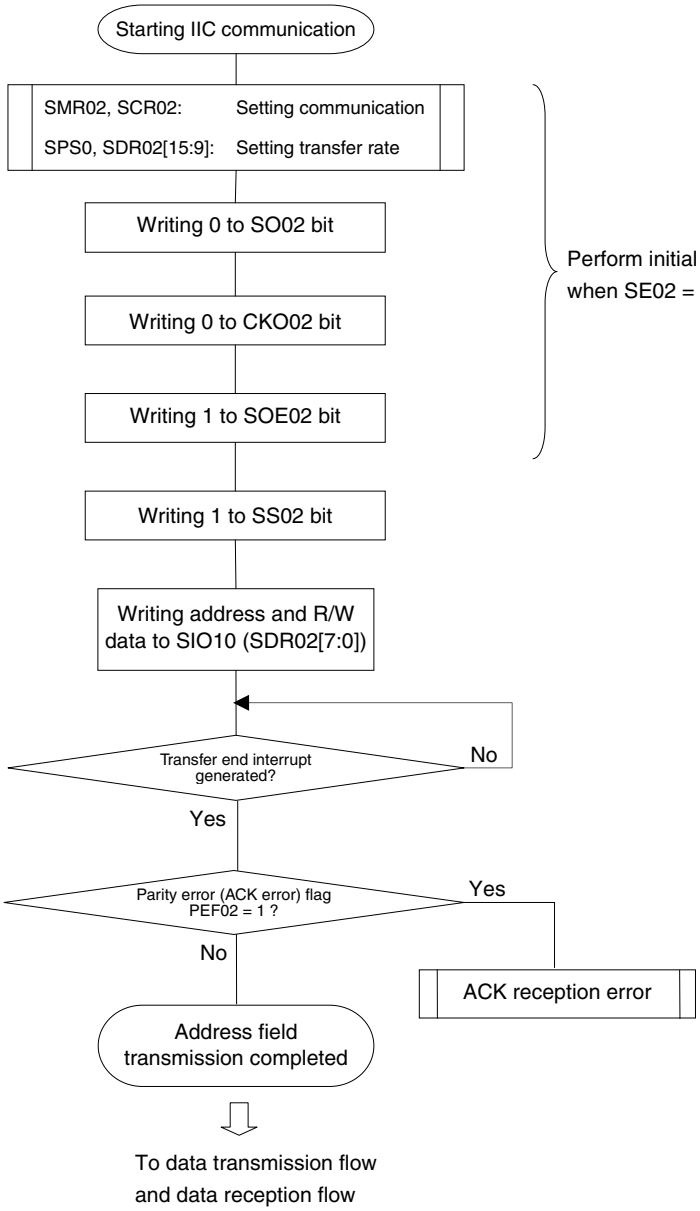


Figure 11-91. Flowchart of Address Field Transmission



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
Target channel	Channel 2 of SAU0
Pins used	SCL10, SDA10
Interrupt	INTIIC10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Parity error detection flag (PEF02)
Transfer data length	8 bits
Transfer rate	Max. f_{CLK}/4 MHz f_{CLK}: System clock frequency 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

(1) Register setting

Figure 11-92. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC10)

(a) Serial output register 0 (SO0) … Do not manipulate this register during data transmission/reception.

<R>

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 0/1Note	1	CKO00 ×	0	0	0	0	1	SO02 0/1Note	1	SO00 ×

(b) Serial output enable register 0 (SOE0) … Do not manipulate this register during data transmission/reception.

<R>

	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 1	0	SOE00 ×

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 ×

(d) Serial mode register 02 (SMR02) … 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
SMR02	CKS02 0/1	CCS02 0	0	0	0	0	0	STS02 0	0	SIS020 0	1	0	0	MD022 1	MD021 0	MD020 0

(e) Serial communication operation setting register 02 (SCR02) … Do not manipulate the bits of this register, except the TXE02 and RXE02 bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR02	TXE02 1	RXE02 0	DAP02 0	CKP02 0	0	EOC02 0	PTC021 0	PTC020 0	DIR02 0	0	SLC021 0	SLC020 1	0	DLS022 1	DLS021 1	DLS020 1

(f) Serial data register 02 (SDR02) (lower 8 bits: SIO10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR02	Baud rate setting							0	Transmit data setting							

SIO10

<R> **Note** The value varies depending on the communication data during communication operation.

Remark □: 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 11-93. Timing Chart of Data Transmission

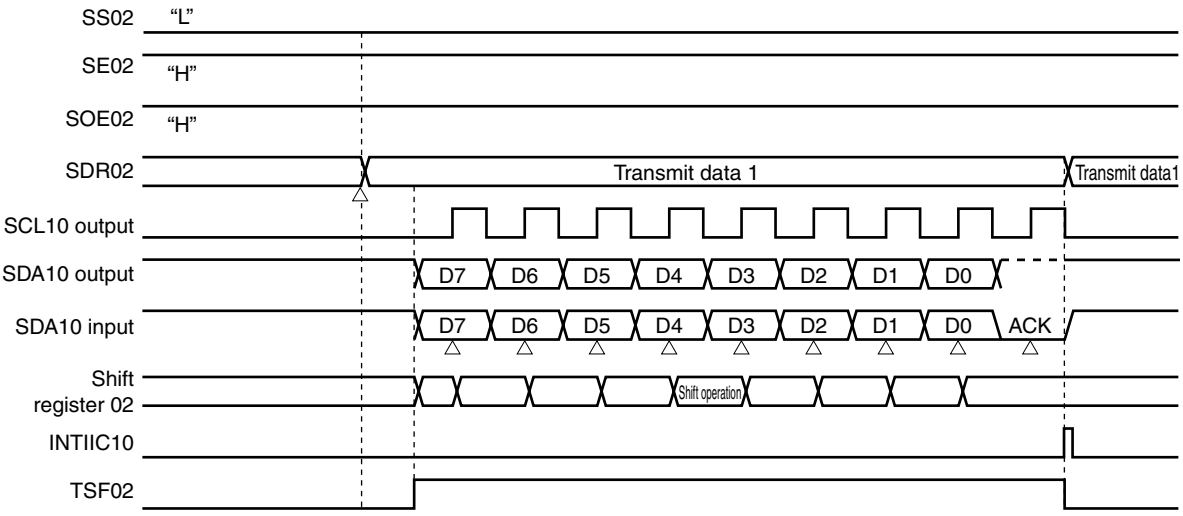
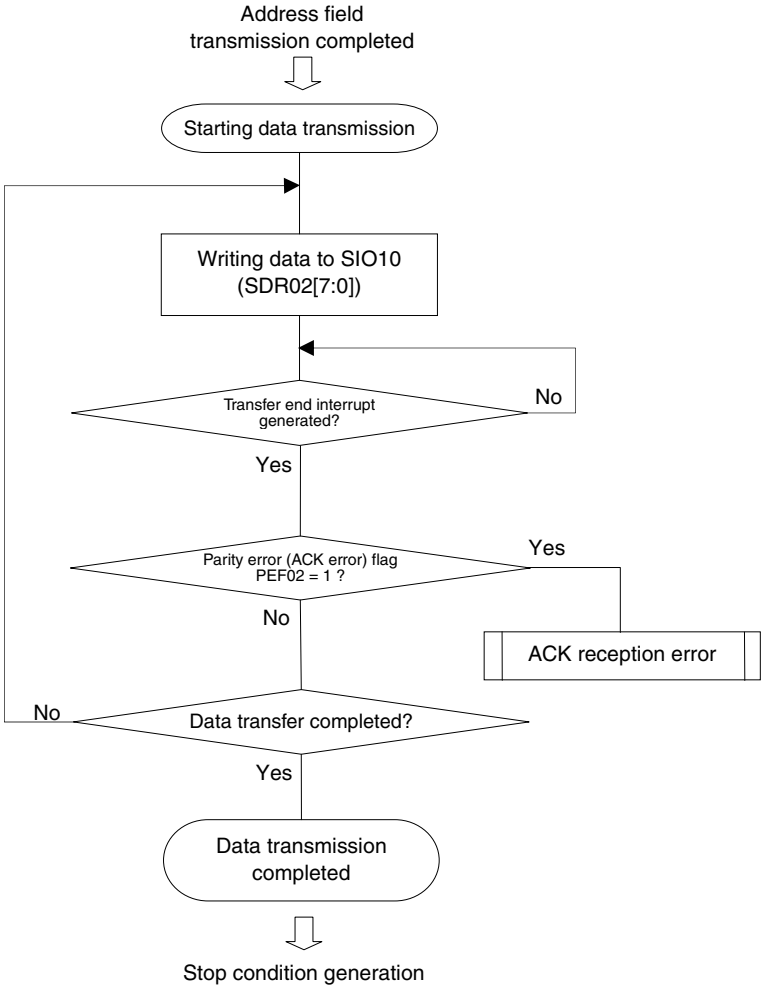


Figure 11-94. Flowchart of Data Transmission



<R>

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
Target channel	Channel 2 of SAU0
Pins used	SCL10, SDA10
Interrupt	INTIIC10
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	None
Transfer data length	8 bits
Transfer rate	Max. f_{CLK}/4 MHz f_{CLK}: System clock frequency 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

(1) Register setting

Figure 11-95. Example of Contents of Registers for Data Reception of Simplified I^2C (IIC10)

(a) Serial output register 0 (SO0) … Do not manipulate this register during data transmission/reception.

<R>

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	CKO02 0/1 Note	1	CKO00 ×	0	0	0	0	1	SO02 0/1 Note	1	SO00 ×

(b) Serial output enable register 0 (SOE0) … Do not manipulate this register during data transmission/reception.

<R>

	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 1	0	SOE00 ×

(c) Serial channel start register 0 (SS0) … 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
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03 ×	SS02 0/1	SS01 ×	SS00 ×

(d) Serial mode register 02 (SMR02) … 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
SMR02	CKS02 0/1	CCS02 0	0	0	0	0	0	STS02 0	0	SIS020 0	1	0	0	MD022 1	MD021 0	MD020 0

(e) Serial communication operation setting register 02 (SCR02) … Do not manipulate the bits of this register, except the TXE02 and RXE02 bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR02	TXE02 0	RXE02 1	DAP02 0	CKP02 0	0	EOC02 0	PTC021 0	PTC020 0	DIR02 0	0	SLC021 0	SLC020 1	0	DLS022 1	DLS021 1	DLS020 1

(f) Serial data register 02 (SDR02) (lower 8 bits: SIO10)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDR02	Baud rate setting							0	Dummy transmit data setting (FFH)							

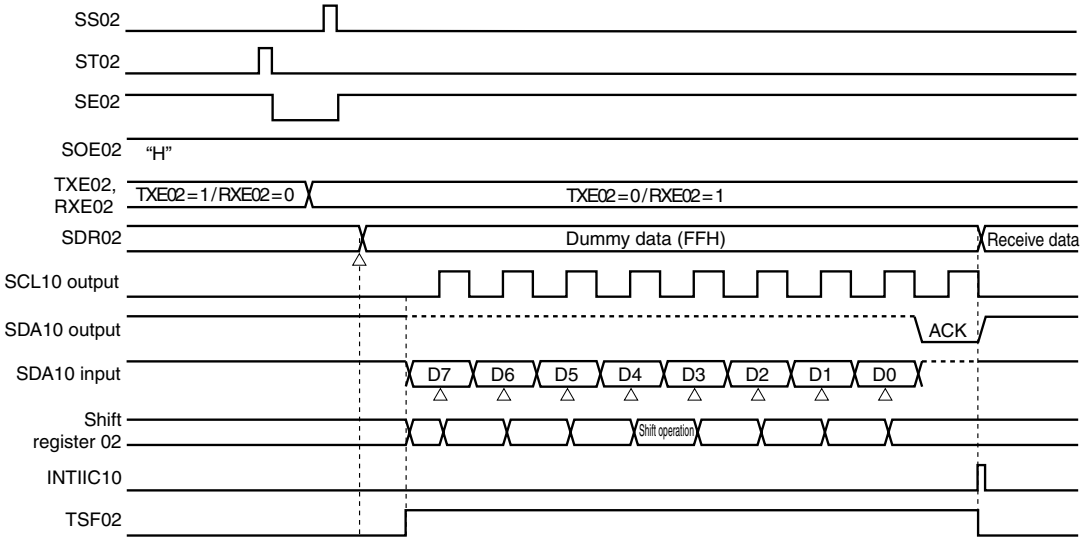
SIO10

<R> **Note** The value varies depending on the communication data during communication operation.

Remark □: 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

<R> **Figure 11-96. Timing Chart of Data Reception**



<R> **Figure 11-97. Flowchart of Data Reception**

<R> **Caution ACK is also output when the last data is received. 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 11-98. Timing Chart of Stop Condition Generation

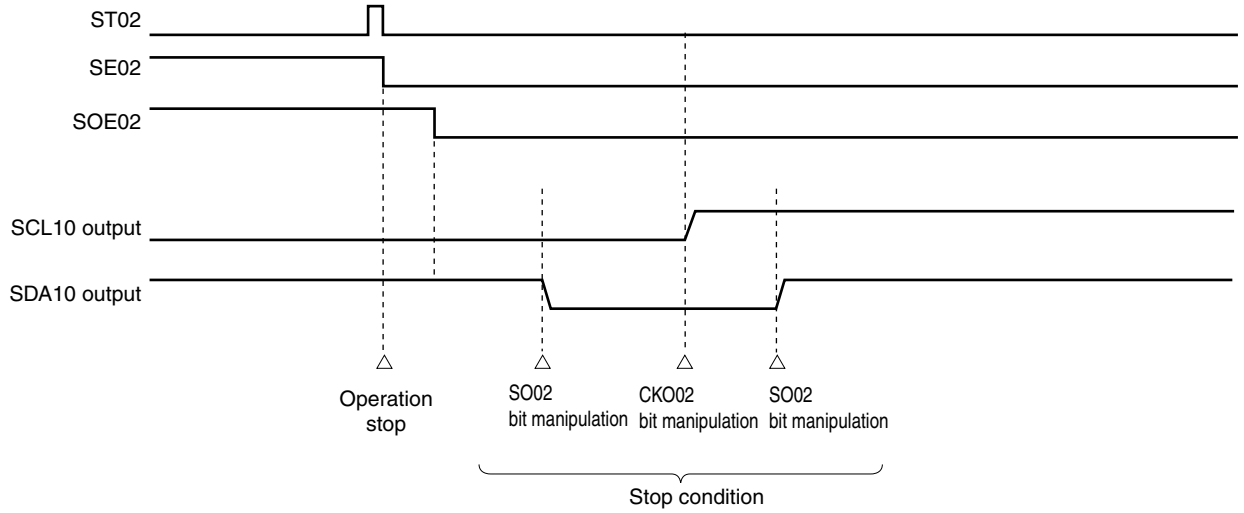
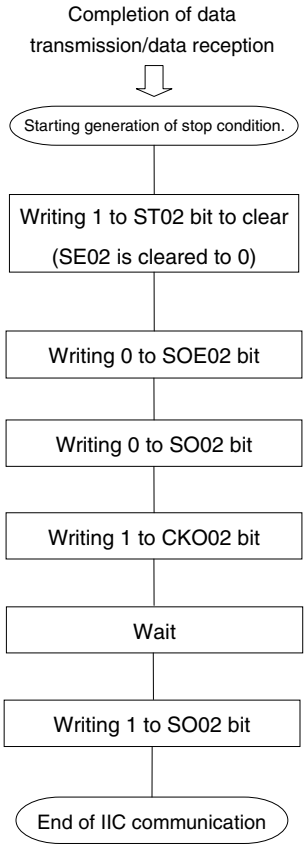


Figure 11-99. Flowchart of Stop Condition Generation



<R>

Secure a wait time so that the specifications of I^2C on the slave side are satisfied.

11.7.5 Calculating transfer rate

The transfer rate for simplified I^2C (IIC10) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (MCK) frequency of target channel} ÷ (SDR02[15:9] + 1) ÷ 2

Remark The value of SDR02[15:9] is the value of bits 15 to 9 of the SDR02 register (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (MCK) is determined by serial clock select register 0 (SPS0) and bit 15 (CKS02) of serial mode register 02 (SMR02).

Table 11-4 Operating Clock Selection

SMR02 Register	SPS0 Register								Operation Clock (MCK) Note 1	
CKS02	PRS 013	PRS 012	PRS 011	PRS 010	PRS 003	PRS 002	PRS 001	PRS 000		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 Note 2	
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 Note 2	
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 (ST0 = 000FH) the operation of the serial array unit (SAU). When selecting INTTM02 for the operation clock, also stop the timer array unit (TAU) (TT0 = 00FFH).

2. SAU can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by setting the TIS02 bit of the TIS0 register of TAU to 1, selecting $f_{SUB}/4$ for the input clock, and selecting INTTM02 using the SPS0 register. When changing f_{CLK}, however, SAU and TAU must be stopped as described in Note 1 above.

Remark X: Don't care

Here is an example of setting an IIC transfer rate where MCK = f_{CLK} = 20 MHz.

IIC Transfer Mode (Desired Transfer Rate)	f_{CLK} = 20 MHz			
	Operation Clock (MCK)	SDR02[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%

11.8 Processing Procedure in Case of Error

The processing procedure to be followed if an error of each type occurs is described in Figures 11-100 to 11-102.

Figure 11-100. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads SDRmn register. →	▶ BFF = 0, and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register. →	▶ Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 11-101. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads SDRmn register. →	▶ BFF = 0, and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register. →	▶ Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets STmn bit to 1. →	▶ SEmn = 0, and channel n stops operation.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets SSmn bit to 1. →	▶ SEmn = 1, and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), 00 to 03, 12, 13

Figure 11-102. Processing Procedure in Case of Parity Error (ACK error) in Simplified I^2C Mode

Software Manipulation	Hardware Status	Remark
Reads SDR02 register. →	BFF = 0, and channel 2 is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads SSR02 register.		Error type is identified and the read value is used to clear error flag.
Writes SIR02 register. →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSR02 register to the SIR02 register without modification.
Sets ST02 bit to 1. →	SE02 = 0, and channel 2 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 SS02 bit to 1. →	SE02 = 1, and channel 2 is enabled to operate.	

<R> 11.9 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 Note1	MD002	MD001	SOE00	SO00	CKO00	TXE00	RXE00	PM10	P10	PM11 Note2	P11 Note2	PM12	P12	Operation mode	Pin Function		
															$\overline{\text{SCK00}}$/P10	SI00/RxD0/P11 Note2	SO00/TxD0/P12
0	0	0	0	1	1	0	0	× Note3	× Note3	× Note3	× Note3	× Note3	× Note3	Operation stop mode	P10	P11	P12
	0	1														P11/RxD0	
1	0	0	0	1	1	0	1	1	×	1	×	× Note3	× Note3	Slave CSI00 reception	$\overline{\text{SCK00}}$ (input)	SI00	P12
			1	0/1 Note4	1	1	0	1	×	× Note3	× Note3	0	1	Slave CSI00 transmission	$\overline{\text{SCK00}}$ (input)	P11	SO00
			1	0/1 Note4	1	1	1	1	×	1	×	0	1	Slave CSI00 transmission/ reception	$\overline{\text{SCK00}}$ (input)	SI00	SO00
			0	1	0/1 Note4	0	1	0	1	1	×	× Note3	× Note3	Master CSI00 reception	$\overline{\text{SCK00}}$ (output)	SI00	P12
			1	0/1 Note4	0/1 Note4	1	0	0	1	× Note3	× Note3	0	1	Master CSI00 transmission	$\overline{\text{SCK00}}$ (output)	P11	SO00
			1	0/1 Note4	0/1 Note4	1	1	0	1	1	×	0	1	Master CSI00 transmission/ reception	$\overline{\text{SCK00}}$ (output)	SI00	SO00
	0	1	1	0/1 Note4	1	1	0	× Note3	× Note3	× Note3	× Note3	0	1	UART0 transmission Note5	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: UART0 reception)

SE01[Note1]	MD012	MD011	TXE01	RXE01	PM11[Note2]	P11[Note2]	Operation mode	Pin Function
								SI00/RxD0/P11[Note2]
0	0	1	0	0	×[Note3]	×[Note3]	Operation stop mode	SI00/P11
1	0	1	0	1	1	×	UART0 reception[Note4, 5]	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.

3. This pin can be set as a port function pin.

4. When using UART0 transmission and reception in a pair, set channel 0 of unit 0 to UART0 transmission (refer to **Table 11-5**).

5. The SMR00 register of channel 0 of unit 0 must also be set during UART0 reception. For details, refer to **11.5.2 (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 Note1	MD022	MD021	SOE02	SO02	CKO02	TXE02	RXE02	PM04	P04	PM03 Note2	P03 Note2	PM02	P02	Operation mode	Pin Function		
															$\overline{SCK10}$/SCL10/P04	SI10/SDA10/RxD1/P03 Note2	SO10/TxD1/P02
0	0	0	0	1	1	0	0	× Note3	× Note3	× Note3	× Note3	× Note3	× Note3	Operation stop mode	P04	P03	P02
	0	1														P03/RxD1	
	1	0														P03	
1	0	0	0	1	1	0	1	1	×	1	×	× Note3	× Note3	Slave CSI10 reception	$\overline{SCK10}$ (input)	SI10	P02
			1	0/1 Note4	1	1	0	1	×	× Note3	× Note3	0	1	Slave CSI10 transmission	$\overline{SCK10}$ (input)	P03	SO10
			1	0/1 Note4	1	1	1	1	×	1	×	0	1	Slave CSI10 transmission /reception	$\overline{SCK10}$ (input)	SI10	SO10
			0	1	0/1 Note4	0	1	0	1	1	×	× Note3	× Note3	Master CSI10 reception	$\overline{SCK10}$ (output)	SI10	P02
			1	0/1 Note4	0/1 Note4	1	0	0	1	× Note3	× Note3	0	1	Master CSI10 transmission	$\overline{SCK10}$ (output)	P03	SO10
			1	0/1 Note4	0/1 Note4	1	1	0	1	1	×	0	1	Master CSI10 transmission /reception	$\overline{SCK10}$ (output)	SI10	SO10
	0	1	1	0/1 Note4	1	1	0	× Note3	× Note3	× Note3	× Note3	0	1	UART1 transmission Note5	P04	P03/RxD1	TxD1
0	1	0	0	0/1 Note6	0/1 Note6	0	0	0	1	0	1	× Note3	× Note3	IIC10 start condition	SCL10	SDA10	P02
						1	0										
						0	1										
1			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	× Note3	× Note3	IIC10 address field transmission	SCL10	SDA10	P02
			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	× Note3	× Note3	IIC10 data transmission	SCL10	SDA10	P02
			1	0/1 Note4	0/1 Note4	0	1	0	1	0	1	× Note3	× Note3	IIC10 data reception	SCL10	SDA10	P02
0			0	0/1 Note7	0/1 Note7	0	0	0	1	0	1	× Note3	× Note3	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 Note1	MD032	MD031	TXE03	RXE03	PM03 Note2	P03 Note2	Operation mode	Pin Function
								SI10/SDA10/RxD1/P03 Note2
0	0	1	0	0	× Note3	× Note3	Operation stop mode	SI10/SDA10/P03 Note2
1	0	1	0	1	1	×	UART1 reception Note4, 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 **11.5.2 (1) Register setting**.

Remark X: Don't care

Table 11-9. Relationship between register settings and pins (Channel 2 of unit 1: UART3 transmission)

SE12 Note1	MD122	MD121	SOE12	SO12	TXE12	RXE12	PM13	P13	Operation mode	Pin Function
										TxD3/P13
0	0	1	0	1	0	0	× Note2	× Note2	Operation stop mode	P13
1	0	1	1	0/1 Note3	1	0	0	1	UART3 transmission Note4	TxD3

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. This pin can be set as a port function pin.

3. This is 0 or 1, depending on the communication operation. For details, refer to **11.3 (12) Serial output register m (SOm)**.

4. When using UART3 transmission and reception in a pair, set channel 3 of unit 1 to UART3 reception (refer to **Table 11-10**).

Remark X: Don't care

Table 11-10. Relationship between register settings and pins (Channel 3 of unit 1: UART3 reception)

SE13 Note1	MD132	MD131	TXE13	RXE13	PM14	P14	Operation mode	Pin Function
								RxD3/P14
0	0	1	0	0	× Note2	× Note2	Operation stop mode	P14
1	0	1	0	1	1	×	UART3 reception Note3, 4	RxD3

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. This pin can be set as a port function pin.

3. When using UART3 transmission and reception in a pair, set channel 2 of unit 1 to UART3 transmission (refer to **Table 11-9**).

4. The SMR12 register of channel 2 of unit 1 must also be set during UART3 reception. For details, refer to **11.5.2 (1) Register setting**.

Remark X: Don't care

CHAPTER 12 SERIAL INTERFACE IIC0

12.1 Functions of Serial Interface IIC0

Serial interface IIC0 has the following two modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^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, IIC0 requires pull-up resistors for the serial clock line and the serial data bus line.

Figure 12-1 shows a block diagram of serial interface IIC0.

Figure 12-1. Block Diagram of Serial Interface IIC0

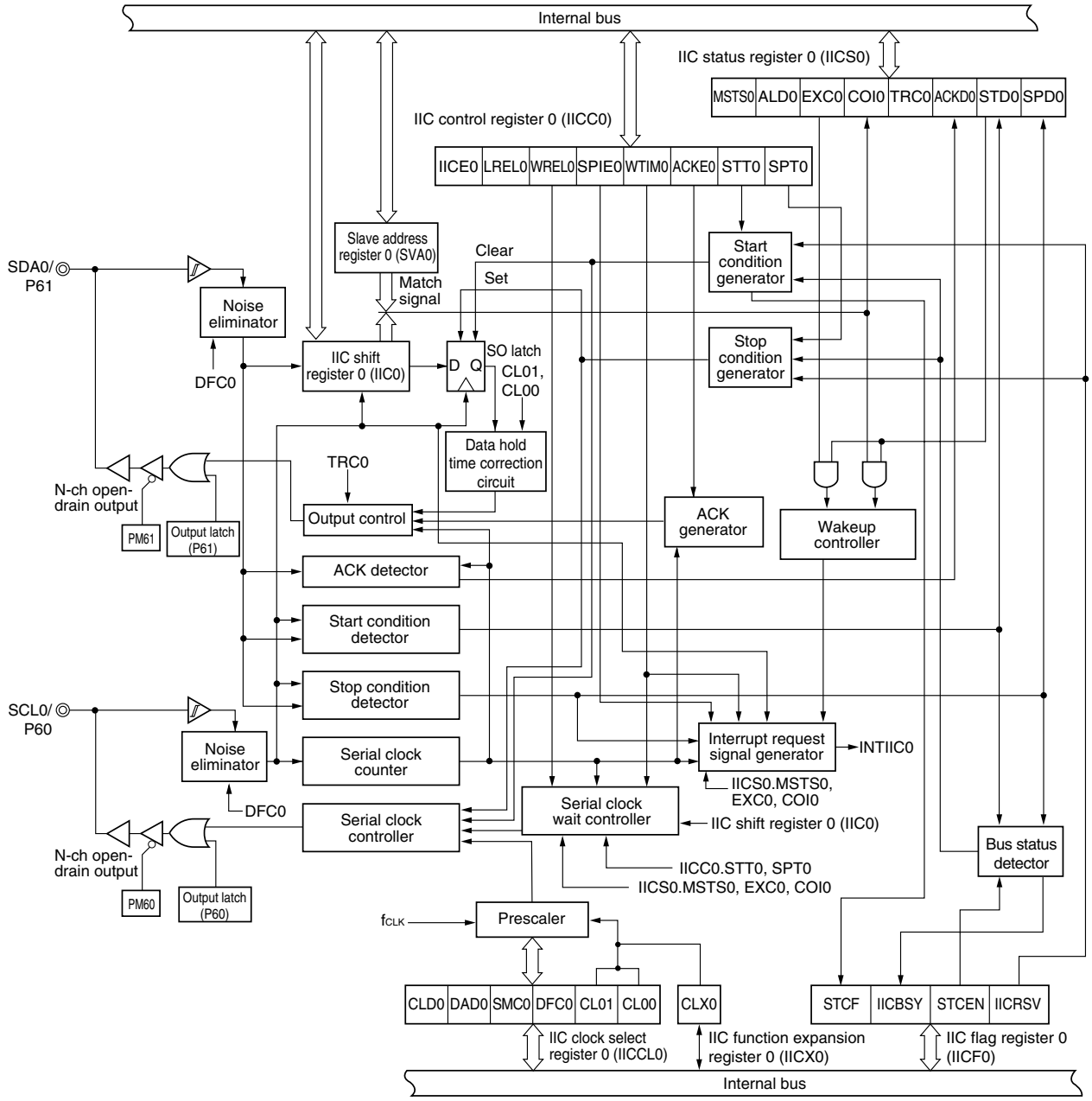
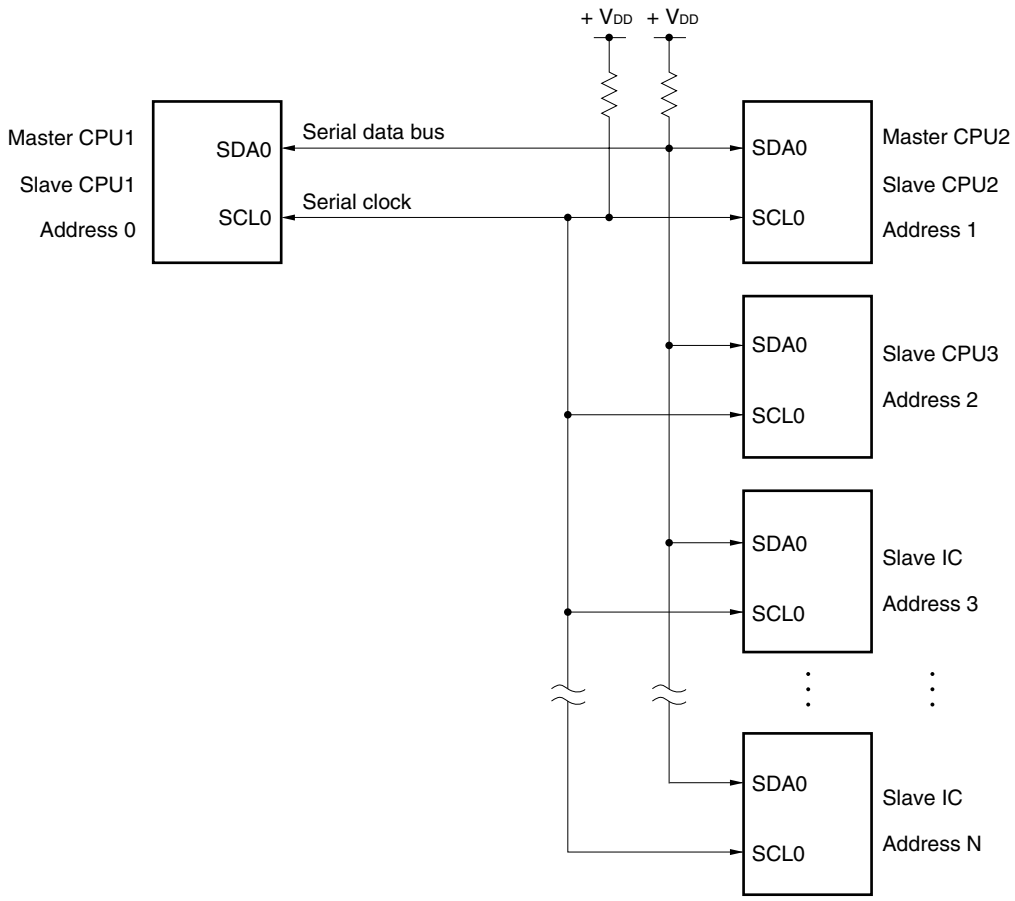


Figure 12-2 shows a serial bus configuration example.

Figure 12-2. Serial Bus Configuration Example Using I^2C Bus



12.2 Configuration of Serial Interface IIC0

Serial interface IIC0 includes the following hardware.

Table 12-1. Configuration of Serial Interface IIC0

Item	Configuration
Registers	IIC shift register 0 (IIC0) Slave address register 0 (SVA0)
Control registers	Peripheral enable register 0 (PER0) IIC control register 0 (IICC0) IIC status register 0 (IICS0) IIC flag register 0 (IICF0) IIC clock select register 0 (IICCL0) IIC function expansion register 0 (IICX0) Port mode register 6 (PM6) Port register 6 (P6)

(1) IIC shift register 0 (IIC0)

IIC0 is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. IIC0 can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to IIC0.

Cancel the wait state and start data transfer by writing data to IIC0 during the wait period.

IIC0 can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears IIC0 to 00H.

Figure 12-3. Format of IIC Shift Register 0 (IIC0)

Address: FFF50H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IIC0								

Cautions 1. Do not write data to IIC0 during data transfer.

2. Write or read IIC0 only during the wait period. Accessing IIC0 in a communication state other than during the wait period is prohibited. When the device serves as the master, however, IIC0 can be written only once after the communication trigger bit (STT0) is set to 1.

(2) Slave address register 0 (SVA0)

This register stores local addresses when in slave mode.

SVA0 can be set by an 8-bit memory manipulation instruction.

However, rewriting to this register is prohibited while STD0 = 1 (while the start condition is detected).

Reset signal generation clears SVA0 to 00H.

Figure 12-4. Format of Slave Address Register 0 (SVA0)

Address: FFF53H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0								0[Note]

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDA0 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIIC0) when the address received by this register matches the address value set to slave address register 0 (SVA0) or when an extension code is received.

(5) Prescaler

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIIC0).

An I^2C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by SPIE0 bit)

Remark WTIM0 bit: Bit 3 of IIC control register 0 (IICC0)
SPIE0 bit: Bit 4 of IIC control register 0 (IICC0)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCL0 pin from a sampling clock.

(9) Serial clock wait controller

This circuit controls the wait timing.

(10) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate and detect each status.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(12) Start condition generator

This circuit generates a start condition when the STT0 bit is set to 1.

However, in the communication reservation disabled status (IICRSV bit = 1), when the bus is not released (IICBSY bit = 1), start condition requests are ignored and the STCF bit is set to 1.

(13) Stop condition generator

This circuit generates a stop condition when the SPT0 bit is set to 1.

(14) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCEN bit.

Remark STT0 bit: Bit 1 of IIC control register 0 (IICC0)
SPT0 bit: Bit 0 of IIC control register 0 (IICC0)
IICRSV bit: Bit 0 of IIC flag register 0 (IICF0)
IICBSY bit: Bit 6 of IIC flag register 0 (IICF0)
STCF bit: Bit 7 of IIC flag register 0 (IICF0)
STCEN bit: Bit 1 of IIC flag register 0 (IICF0)

12.3 Registers to Controlling Serial Interface IIC0

Serial interface IIC0 is controlled by the following eight registers.

- Peripheral enable register 0 (PER0)
- IIC control register 0 (IICC0)
- IIC flag register 0 (IICF0)
- IIC status register 0 (IICS0)
- IIC clock select register 0 (IICCL0)
- IIC function expansion register 0 (IICX0)
- Port mode register 6 (PM6)
- Port register 6 (P6)

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial interface IIC0 is used, be sure to set bit 4 (IIC0EN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Cautions 1. When setting serial interface IIC0, be sure to set IIC0EN to 1 first. If IIC0EN = 0, writing to a control register of serial interface IIC0 is ignored, and, even if the register is read, only the default value is read.

2. Be sure to clear bits 1 and 6 of PER0 register to 0.

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	IIC0EN	SAU1EN	SAU0EN	0	TAU0EN

IIC0EN	Control of serial interface IIC0 input clock
0	Stops supply of input clock. • SFR used by serial interface IIC0 cannot be written. • Serial interface IIC0 is in the reset status.
1	Supplies input clock. • SFR used by serial interface IIC0 can be read/written.

(2) IIC control register 0 (IICC0)

This register is used to enable/stop I^2C operations, set wait timing, and set other I^2C operations.

IICC0 can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE0, WTIM0, and ACKE0 bits while IICE0 bit = 0 or during the wait period. These bits can be set at the same time when the IICE0 bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Figure 12-6. Format of IIC Control Register 0 (IICC0) (1/4)

Address: FFF52H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC0	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable
0	Stop operation. Reset IIC status register 0 (IICS0)[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 (IICE0 = 0)	Condition for setting (IICE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL0[Note 2]	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL0 and SDA0 lines are set to high impedance. The following flags of IIC control register 0 (IICC0) and IIC status register 0 (IICS0) are cleared to 0. • STT0 • SPT0 • MSTS0 • EXC0 • COI0 • TRC0 • ACKD0 • STD0
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • 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 (LREL0 = 0)	Condition for setting (LREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL0[Note 2]	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When WREL0 is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC0 = 1), the SDA0 line goes into the high impedance state (TRC0 = 0).	
Condition for clearing (WREL0 = 0)	Condition for setting (WREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICS0 register, the STCF0 and IICBSY bits of the IICF0 register, and the CLD0 and DAD0 bits of the IICCL0 register are reset.

2. This flag's signal is invalid when IICE0 = 0.

Caution The start condition is detected immediately after I^2C is enabled to operate (IICE0 = 1) while the SCL0 line is at high level and the SDA0 line is at low level. Immediately after enabling I^2C to operate (IICE0 = 1), set LREL0 (1) by using a 1-bit memory manipulation instruction.

Figure 12-6. Format of IIC Control Register 0 (IICC0) (2/4)

SPIE0[Note 1]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable

Condition for clearing (SPIE0 = 0)	Condition for setting (SPIE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM0[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 (WTIM0 = 0)	Condition for setting (WTIM0 = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE0[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 (ACKE0 = 0)	Condition for setting (ACKE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

Notes 1. This flag's signal is invalid when IICE0 = 0.

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 IIC Control Register 0 (IICC0) (3/4)

STT0[Note]	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in STOP mode): Generate a start condition (for starting as master). When the SCL0 line is high level, the SDA0 line is changed from high level to low level and then the start condition is generated. Next, after the rated amount of time has elapsed, SCL0 is changed to low level (wait state). 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) STCF is set to 1 and information that is set (1) to STT0 is cleared. 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 ACKE0 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 SPT0.
- Setting STT0 to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (STT0 = 0)	Condition for setting (STT0 = 1)
• Cleared by setting SST0 to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Note This flag's signal is invalid when IICE0 = 0.

Remarks 1. Bit 1 (STT0) becomes 0 when it is read after data setting.

2. IICRSV: Bit 0 of IIC flag register (IICF0)
STCF: Bit 7 of IIC flag register (IICF0)

Figure 12-6. Format of IIC Control Register 0 (IICC0) (4/4)

SPT0	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer). After the SDA0 line goes to low level, either set the SCL0 line to high level or wait until it goes to high level. Next, after the rated amount of time has elapsed, the SDA0 line changes from low level to high level and a stop condition is generated.

Cautions concerning set timing

- For master reception: Cannot be set to 1 during transfer.
  Can be set to 1 only in the waiting period when ACKE0 has been cleared to 0 and slave has been notified of final reception.
- 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 STT0.
- SPT0 can be set to 1 only when in master mode[Note].
- When WTIM0 has been cleared to 0, if SPT0 is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. WTIM0 should be changed from 0 to 1 during the wait period following the output of eight clocks, and SPT0 should be set to 1 during the wait period that follows the output of the ninth clock.
- Setting SPT0 to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (SPT0 = 0)	Condition for setting (SPT0 = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Note Set SPT0 to 1 only in master mode. However, SPT0 must be set to 1 and a stop condition generated before the first stop condition is detected following the switch to the operation enabled status.

Caution When bit 3 (TRC0) of IIC status register 0 (IICS0) is set to 1, WREL0 is set to 1 during the ninth clock and wait is canceled, after which TRC0 is cleared and the SDA0 line is set to high impedance.

Remark Bit 0 (SPT0) becomes 0 when it is read after data setting.

(3) IIC status register 0 (IICS0)

This register indicates the status of I^2C.

IICS0 is read by a 1-bit or 8-bit memory manipulation instruction only when STT0 = 1 and during the wait period.

Reset signal generation clears this register to 00H.

Figure 12-7. Format of IIC Status Register 0 (IICS0) (1/3)

Address: FFF56H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master device status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS0 = 0)	Condition for setting (MSTS0 = 1)
• When a stop condition is detected • When ALD0 = 1 (arbitration loss) • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD0	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. MSTS0 is cleared.

Condition for clearing (ALD0 = 0)	Condition for setting (ALD0 = 1)
• Automatically cleared after IICS0 is read[Note] • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0 = 0)	Condition for setting (EXC0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 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).

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than IICS0. Therefore, when using the ALD0 bit, read the data of this bit before the data of the other bits.

Remark LREL0: Bit 6 of IIC control register 0 (IICC0)
IICE0: Bit 7 of IIC control register 0 (IICC0)

Figure 12-7. Format of IIC Status Register 0 (IICS0) (2/3)

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI0 = 0)	Condition for setting (COI0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register 0 (SVA0)) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDA0 line (valid starting at the falling edge of the first byte's ninth clock).

Condition for clearing (TRC0 = 0)	Condition for setting (TRC0 = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Cleared by WREL0 = 1[Note] (wait cancel) • When ALD0 changes from 0 to 1 (arbitration loss) • Reset <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) <When not used for communication>	<Master> • When a start condition is generated • When "0" is output to the first byte's LSB (transfer direction specification bit) <Slave> • When "1" is input to the first byte's LSB (transfer direction specification bit)

Note If the wait state is canceled by setting bit 5 (WREL0) of IIC control register 0 (IICC0) to 1 at the ninth clock when bit 3 (TRC0) of IIC status register 0 (IICS0) is 1, TRC0 is cleared, and the SDA0 line goes into a high-impedance state.

Remark LREL0: Bit 6 of IIC control register 0 (IICC0)
IICE0: Bit 7 of IIC control register 0 (IICC0)

Figure 12-7. Format of IIC Status Register 0 (IICS0) (3/3)

ACKD0	Detection of acknowledge ($\overline{ACK}$)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKD0 = 0)	Condition for setting (ACKD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• After the SDA0 line is set to low level at the rising edge of SCL0's ninth clock

STD0	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect.

Condition for clearing (STD0 = 0)	Condition for setting (STD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL0 = 1 (exit from communications) • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD0	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.

Condition for clearing (SPD0 = 0)	Condition for setting (SPD0 = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When IICE0 changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark LREL0: Bit 6 of IIC control register 0 (IICC0)
IICE0: Bit 7 of IIC control register 0 (IICC0)

(4) IIC flag register 0 (IICF0)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.

IICF0 can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STCF and IICBSY bits are read-only.

The IICRSV bit can be used to enable/disable the communication reservation function.

STCEN can be used to set the initial value of the IICBSY bit.

IICRSV and STCEN can be written only when the operation of I^2C is disabled (bit 7 (IICE0) of IIC control register 0 (IICC0) = 0). When operation is enabled, the IICF0 register can be read.

Reset signal generation clears this register to 00H.

Figure 12-8. Format of IIC Flag Register 0 (IICF0)

Address: FFF51H After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	STT0 clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear STT0 flag

Condition for clearing (STCF = 0)	Condition for setting (STCF = 1)
• Cleared by STT0 = 1 • When IICE0 = 0 (operation stop) • Reset	• Generating start condition unsuccessful and STT0 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 IICE0 = 0 (operation stop) • Reset	• Detection of start condition • Setting of IICE0 when STCEN = 0

STCEN	Initial start enable trigger
0	After operation is enabled (IICE0 = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE0 = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN = 0)	Condition for setting (STCEN = 1)
• 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 STCEN only when the operation is stopped (IICE0 = 0).

2. As the bus release status (IICBSY = 0) is recognized regardless of the actual bus status when STCEN = 1, when generating the first start condition (STT0 = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

3. Write to IICRSV only when the operation is stopped (IICE0 = 0).

Remark STT0: Bit 1 of IIC control register 0 (IICC0)
IICE0: Bit 7 of IIC control register 0 (IICC0)

(5) IIC clock select register 0 (IICCL0)

This register is used to set the transfer clock for the I^2C bus.

IICCL0 can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD0 and DAD0 bits are read-only. The SMC0, CL01, and CL00 bits are set in combination with bit 0 (CLX0) of IIC function expansion register 0 (IICX0) (see **12.5.4 Transfer clock setting method**).

Set IICCL0 while bit 7 (IICE0) of IIC control register 0 (IICC0) is 0.

Reset signal generation clears this register to 00H.

Figure 12-9. Format of IIC Clock Select Register 0 (IICCL0)

Address: FFF54H After reset: 00H R/W[Note]

Symbol	7	6	<5>	<4>	<3>	<2>	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	CL01	CL00

CLD0	Detection of SCL0 pin level (valid only when IICE0 = 1)
0	The SCL0 pin was detected at low level.
1	The SCL0 pin was detected at high level.

Condition for clearing (CLD0 = 0)	Condition for setting (CLD0 = 1)
• When the SCL0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SCL0 pin is at high level

DAD0	Detection of SDA0 pin level (valid only when IICE0 = 1)
0	The SDA0 pin was detected at low level.
1	The SDA0 pin was detected at high level.

Condition for clearing (DAD0 = 0)	Condition for setting (DAD0 = 1)
• When the SDA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SDA0 pin is at high level

SMC0	Operation mode switching
0	Operates in standard mode.
1	Operates in fast mode.

DFC0	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in fast mode. In fast mode, the transfer clock does not vary regardless of DFC0 bit set (1)/clear (0). The digital filter is used for noise elimination in fast mode.	

Note Bits 4 and 5 are read-only.

Remark IICE0: Bit 7 of IIC control register 0 (IICC0)

(6) IIC function expansion register 0 (IICX0)

This register sets the function expansion of I^2C.

IICX0 can be set by a 1-bit or 8-bit memory manipulation instruction. The CLX0 bit is set in combination with bits 3, 1, and 0 (SMC0, CL01, and CL00) of IIC clock select register 0 (IICCL0) (see **12.5.4 Transfer clock setting method**).

Set IICX0 while bit 7 (IICE0) of IIC control register 0 (IICC0) is 0.

Reset signal generation clears this register to 00H.

Figure 12-10. Format of IIC Function Expansion Register 0 (IICX0)

Address: FFF55H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
IICX0	0	0	0	0	0	0	0	CLX0

Table 12-2. Selection Clock Setting

IICX0	IICCL0			Transfer Clock (f_{CLK}/m)	Settable Selection Clock (f_{CLK}) Range	Operation Mode
Bit 0	Bit 3	Bit 1	Bit 0			
CLX0	SMC0	CL01	CL00			
0	0	0	0	f_{CLK}/88	4.00 MHz to 8.38 MHz	Normal mode (SMC0 bit = 0)
0	0	0	1	f_{CLK}/172	8.38 MHz to 16.76 MHz	
0	0	1	0	f_{CLK}/344	16.76 MHz to 20 MHz	
0	0	1	1	f_{CLK}/44	2.00 MHz to 4.19 MHz	
0	1	0	×	f_{CLK}/48	8.00 MHz to 16.76 MHz	Fast mode (SMC0 bit = 1)
0	1	1	0	f_{CLK}/96	16.00 MHz to 20 MHz	
0	1	1	1	f_{CLK}/24	4.00 MHz to 8.38 MHz	
1	0	×	×	Setting prohibited		
1	1	0	×	f_{CLK}/48	8.00 MHz to 8.38 MHz	Fast mode (SMC0 bit = 1)
1	1	1	0	Setting prohibited	16.00 MHz to 16.76 MHz	
1	1	1	1	f_{CLK}/24	4.00 MHz to 4.19 MHz	

Caution Determine the transfer clock frequency of I^2C by using CLX0, SMC0, CL01, and CL00 before enabling the operation (by setting bit 7 (IICE0) of IIC control register 0 (IICC0) to 1). To change the transfer clock frequency, clear IICE0 once to 0.

Remarks 1. ×: don't care

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

(7) Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCL0 pin as clock I/O and the P61/SDA0 pin as serial data I/O, clear PM60 and PM61, and the output latches of P60 and P61 to 0.

Set IICE0 (bit 7 of IIC control register 0 (IICC0)) to 1 before setting the output mode because the P60/SCL0 and P61/SDA0 pins output a low level (fixed) when IICE0 is 0.

PM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 12-11. Format of Port Mode Register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	PM63	PM62	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

12.4 I²C Bus Mode Functions

12.4.1 Pin configuration

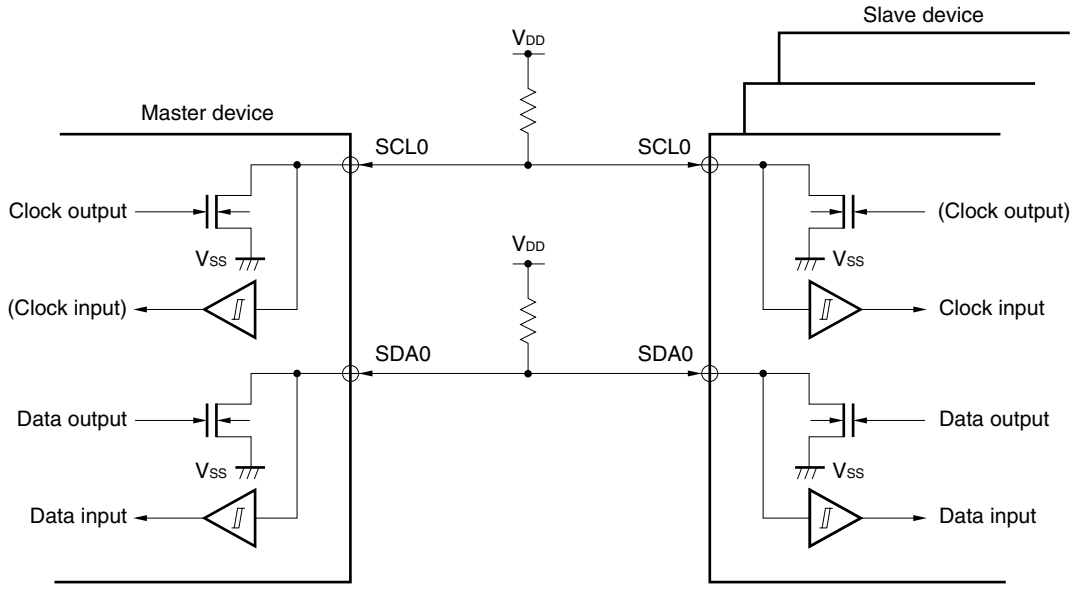
The serial clock pin (SCL0) and serial data bus pin (SDA0) are configured as follows.

(1) SCL0..... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDA0 ..... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

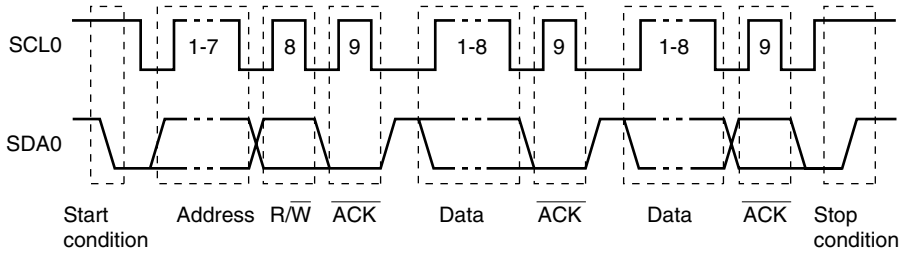
Figure 12-12. Pin Configuration Diagram



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 12-13 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 12-13. I²C Bus Serial Data Transfer Timing



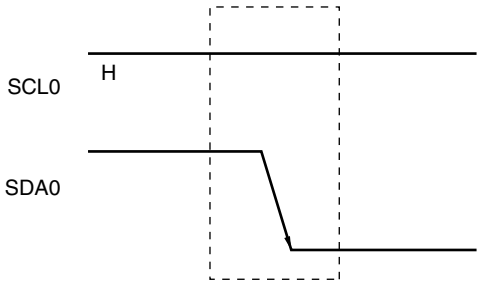
The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCL0) is continuously output by the master device. However, in the slave device, the SCL0's 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 12-14. Start Conditions



A start condition is output when bit 1 (STT0) of IIC control register 0 (IICC0) is set (to 1) after a stop condition has been detected (SPD0: Bit 0 = 1 in IIC status register 0 (IICS0)). When a start condition is detected, bit 1 (STD0) of IICS0 is set (to 1).

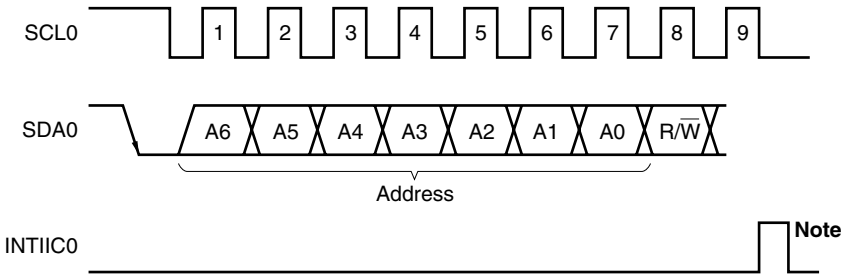
12.5.2 Addresses

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 slave address register 0 (SVA0). If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 12-15. Address



Note INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

The slave address and the eighth bit, which specifies the transfer direction as described in **12.5.3 Transfer direction specification** below, are together written to IIC shift register 0 (IIC0) and are then output. Received addresses are written to IIC0.

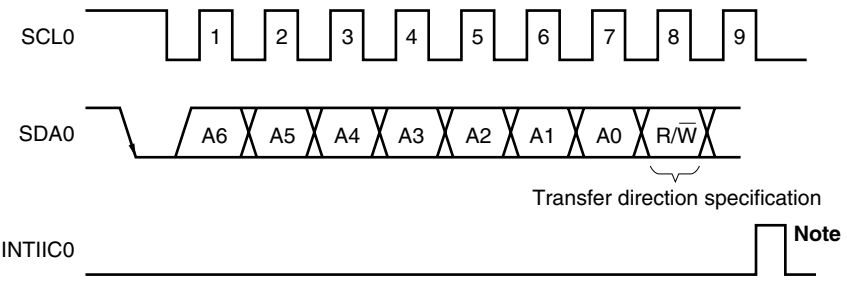
The slave address is assigned to the higher 7 bits of IIC0.

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 12-16. Transfer Direction Specification



Note INTIIC0 is not issued if data other than a local address or extension code is received during slave device operation.

12.5.4 Transfer clock setting method

(1) Selection clock setting method on the master side

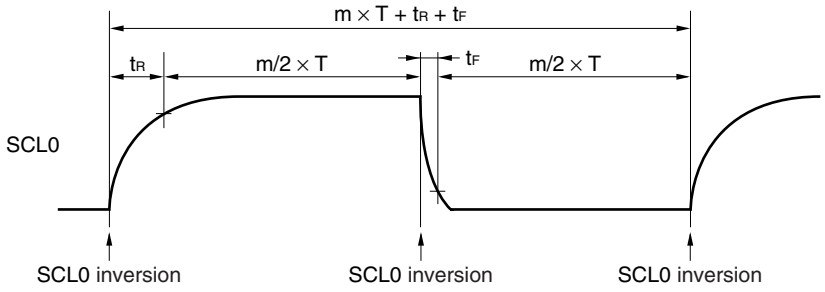
The I^2C transfer clock frequency (f_{SCL}) is calculated using the following expression.

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

m = 24, 44, 48, 88, 96, 172, 344 (see **Table 12-3 Selection Clock Setting**)
T: $1/f_{CLK}$
t_R: SCL0 rise time
t_F: SCL0 fall time

For example, the I^2C transfer clock frequency (f_{SCL}) when f_{CLK} = 4.19 MHz, m = 88, t_R = 200 ns, and t_F = 50 ns is calculated using following expression.

$$f_{SCL} = 1/(88 \times 238.7 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 47.0 \text{ kHz}$$



The selection clock is set using a combination of bits 3, 1, and 0 (SMC0, CL01, and CL00) of IIC clock select register 0 (IICCL0) and bit 0 (CLX0) of IIC function expansion register 0 (IICX0).

(2) Selection clock setting method on the slave side

To use as slave, set the bits 3, 1, and 0 (SMC0, CL01, CL00) of the IIC clock selection register (IICL0) and the bit 0 (CLX0) of the IIC function expansion register 0 (IICX0) according to the f_{CLK} (Selectable Selection Clock Range) and IIC Operation Mode (Normal or Fast) as defined in **Table 12-3. Selection Clock Setting**.

Table 12-3. Selection Clock Setting

IICX0	IICCL0			Transfer Clock (f_{CLK}/m)	Settable Selection Clock (f_{CLK}) Range	Operation Mode
Bit 0	Bit 3	Bit 1	Bit 0			
CLX0	SMC0	CL01	CL00			
0	0	0	0	f_{CLK}/88	4.00 MHz to 8.38 MHz	Normal mode (SMC0 bit = 0)
0	0	0	1	f_{CLK}/172	8.38 MHz to 16.76 MHz	
0	0	1	0	f_{CLK}/344	16.76 MHz to 20 MHz	
0	0	1	1	f_{CLK}/44	2.00 MHz to 4.19 MHz	
0	1	0	×	f_{CLK}/48	8.00 MHz to 16.76 MHz	Fast mode (SMC0 bit = 1)
0	1	1	0	f_{CLK}/96	16.00 MHz to 20 MHz	
0	1	1	1	f_{CLK}/24	4.00 MHz to 8.38 MHz	
1	0	×	×	Setting prohibited		
1	1	0	×	f_{CLK}/48	8.00 MHz to 8.38 MHz	Fast mode (SMC0 bit = 1)
1	1	1	0	Setting prohibited	16.00 MHz to 16.76 MHz	
1	1	1	1	f_{CLK}/24	4.00 MHz to 4.19 MHz	

Caution Determine the transfer clock frequency of I^2C by using CLX0, SMC0, CL01, and CL00 before enabling the operation (by setting bit 7 (IICE0) of IIC control register 0 (IICC0) to 1). To change the transfer clock frequency, clear IICE0 once to 0.

Remarks 1. ×: don't care

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

12.5.5 Acknowledge ($\overline{ACK}$)

$\overline{ACK}$ is used to check the status of serial data at the transmission and reception sides.

The reception side returns $\overline{ACK}$ each time it has received 8-bit data.

The transmission side usually receives $\overline{ACK}$ after transmitting 8-bit data. When $\overline{ACK}$ is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether $\overline{ACK}$ has been detected can be checked by using bit 2 (ACKD0) of IIC status register 0 (IICS0).

When the master receives the last data item, it does not return $\overline{ACK}$ and instead generates a stop condition. If a slave does not return $\overline{ACK}$ after receiving data, the master outputs a stop condition or restart condition and stops transmission. If $\overline{ACK}$ is not returned, the possible causes are as follows.

<1> Reception was not performed normally.

<2> The final data item was received.

<3> The reception side specified by the address does not exist.

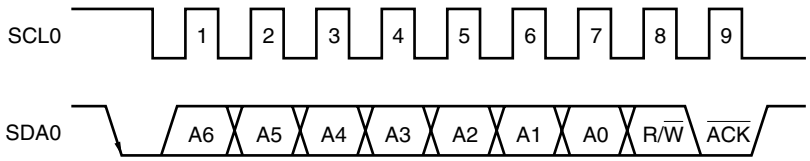
To generate $\overline{ACK}$, the reception side makes the SDA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{ACK}$ is enabled by setting bit 2 (ACKE0) of IIC control register 0 (IICC0) to 1. Bit 3 (TRC0) of the IICS0 register is set by the data of the eighth bit that follows 7-bit address information. Usually, set ACKE0 to 1 for reception (TRC0 = 0).

If a slave can receive no more data during reception (TRC0 = 0) or does not require the next data item, then the slave must inform the master, by clearing ACKE0 to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC0 = 0), it must clear ACKE0 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 12-17. $\overline{\text{ACK}}$



When the local address is received, $\overline{\text{ACK}}$ is automatically generated, regardless of the value of ACKE0. When an address other than that of the local address is received, $\overline{\text{ACK}}$ is not generated (NACK).

When an extension code is received, $\overline{\text{ACK}}$ is generated if ACKE0 is set to 1 in advance.

How $\overline{\text{ACK}}$ is generated when data is received differs as follows depending on the setting of the wait timing.

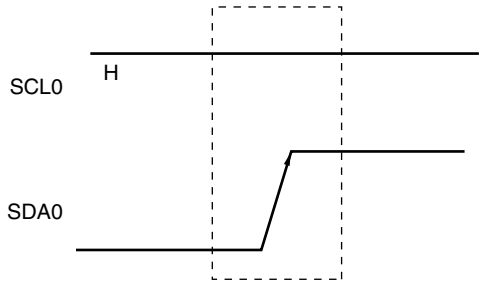
- When 8-clock wait state is selected (bit 3 (WTIM0) of IICC0 register = 0):
  By setting ACKE0 to 1 before releasing the wait state, $\overline{\text{ACK}}$ is generated at the falling edge of the eighth clock of the SCL0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM0) of IICC0 register = 1):
  $\overline{\text{ACK}}$ is generated by setting ACKE0 to 1 in advance.

12.5.6 Stop condition

When the SCL0 pin is at high level, changing the SDA0 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 12-18. Stop Condition



A stop condition is generated when bit 0 (SPT0) of IIC control register 0 (IICC0) is set to 1. When the stop condition is detected, bit 0 (SPD0) of IIC status register 0 (IICS0) is set to 1 and INTIIC0 is generated when bit 4 (SPIE0) of IICC0 is set to 1.

12.5.7 Wait

The wait is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL0 pin to low level notifies the communication partner of the wait state. When wait state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 12-19. Wait (1/2)

(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKE0 = 1)

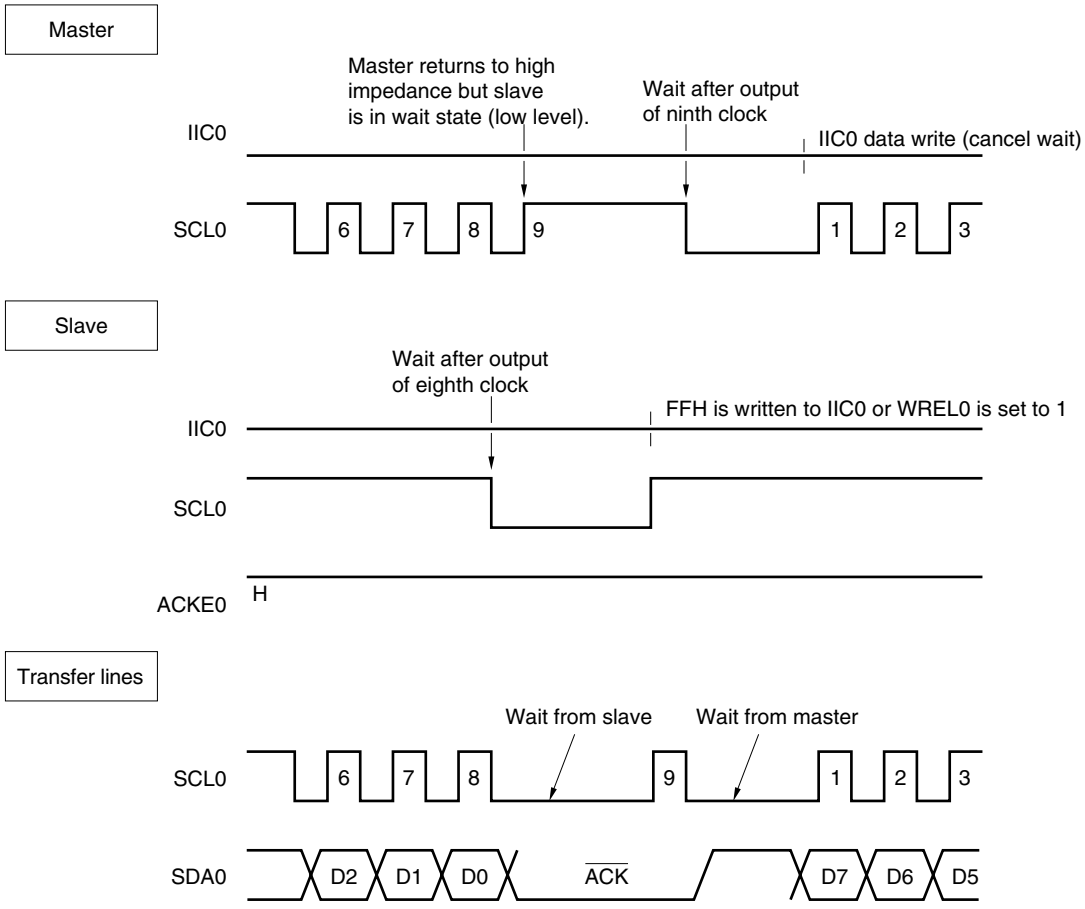
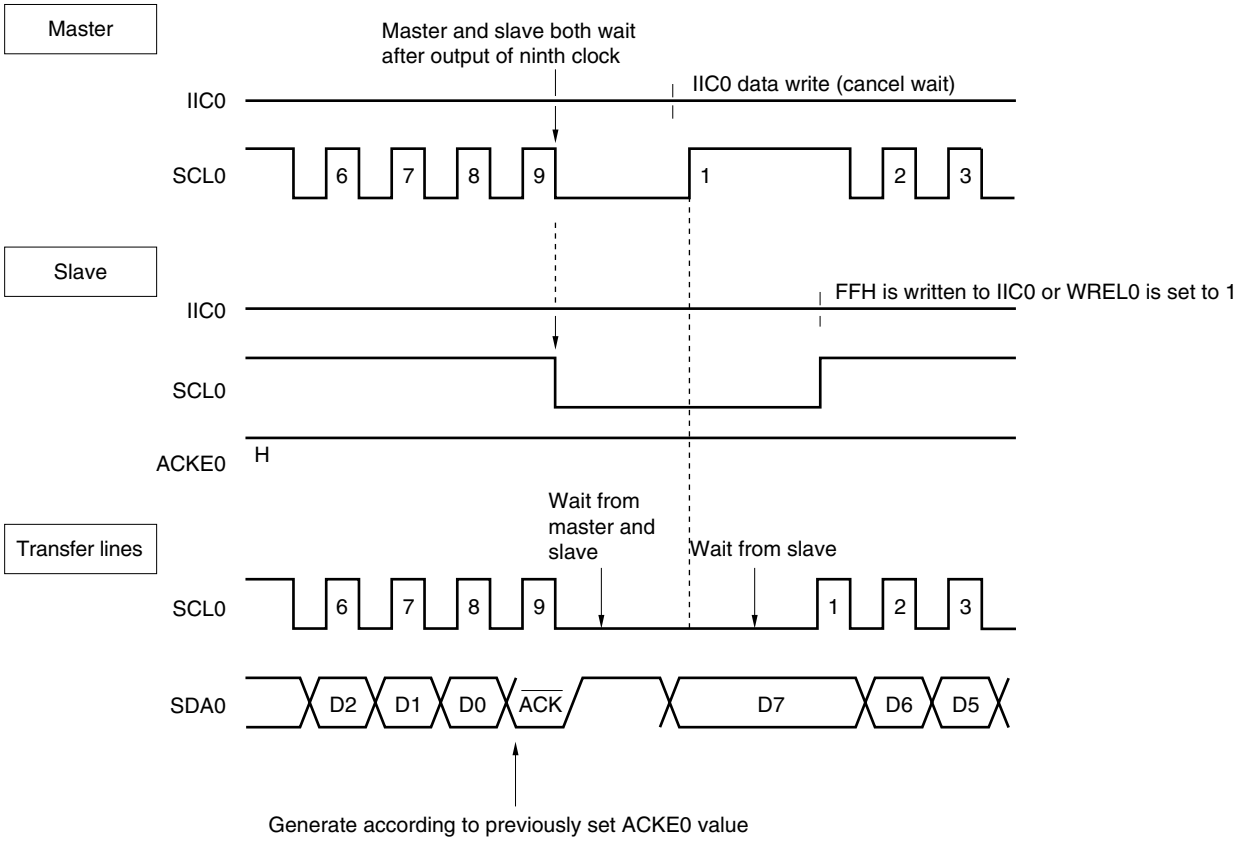


Figure 12-19. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE0 = 1)



Remark ACKE0: Bit 2 of IIC control register 0 (IICC0)
WREL0: Bit 5 of IIC control register 0 (IICC0)

A wait may be automatically generated depending on the setting of bit 3 (WTIM0) of IIC control register 0 (IICC0).

Normally, the receiving side cancels the wait state when bit 5 (WREL0) of IICC0 is set to 1 or when FFH is written to IIC shift register 0 (IIC0), and the transmitting side cancels the wait state when data is written to IIC0.

The master device can also cancel the wait state via either of the following methods.

- By setting bit 1 (STT0) of IICC0 to 1
- By setting bit 0 (SPT0) of IICC0 to 1

12.5.8 Canceling wait

The I^2C usually cancels a wait state by the following processing.

- Writing data to IIC shift register 0 (IIC0)
- Setting bit 5 (WREL0) of IIC control register 0 (IICC0) (canceling wait)
- Setting bit 1 (STT0) of IIC0 register (generating start condition)[Note]
- Setting bit 0 (SPT0) of IIC0 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 IIC0.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL0) of the IIC0 control register 0 (IICC0) to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT0) of IICC0 to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT0) of IICC0 to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to IIC0 after canceling a wait state by setting WREL0 to 1, an incorrect value may be output to SDA0 because the timing for changing the SDA0 line conflicts with the timing for writing IIC0.

In addition to the above, communication is stopped if IICE0 is cleared to 0 when communication has been aborted, so that the wait state can be canceled.

If the I^2C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LREL0) of IICC0, so that the wait state can be canceled.

12.5.9 Interrupt request (INTIIC0) generation timing and wait control

The setting of bit 3 (WTIM0) of IIC control register 0 (IICC0) determines the timing by which INTIIC0 is generated and the corresponding wait control, as shown in Table 12-4.

Table 12-4. INTIIC0 Generation Timing and Wait Control

WTIM0	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	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 INTIIC0 signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to slave address register 0 (SVA0).
At this point, $\overline{ACK}$ is generated regardless of the value set to IICC0's bit 2 (ACKE0). For a slave device that has received an extension code, INTIIC0 occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIIC0 is generated at the falling edge of the 9th clock, but wait does not occur.

2. If the received address does not match the contents of slave address register 0 (SVA0) and extension code is not received, neither INTIIC0 nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIM0 bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to IIC shift register 0 (IIC0)
- Setting bit 5 (WREL0) of IIC control register 0 (IICC0) (canceling wait)
- Setting bit 1 (STT0) of IIC0 register (generating start condition)Note
- Setting bit 0 (SPT0) of IIC0 register (generating stop condition)Note

Note Master only.

When an 8-clock wait has been selected (WTIM0 = 0), the presence/absence of $\overline{ACK}$ generation must be determined prior to wait cancellation.

(5) Stop condition detection

INTIIC0 is generated when a stop condition is detected (only when SPIE0 = 1).

12.5.10 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 (INTIIC0) occurs when a local address has been set to slave address register 0 (SVA0) and when the address set to SVA0 matches the slave address sent by the master device, or when an extension code has been received.

12.5.11 Error detection

In I^2C bus mode, the status of the serial data bus (SDA0) during data transmission is captured by IIC shift register 0 (IIC0) of the transmitting device, so the IIC0 data prior to transmission can be compared with the transmitted IIC0 data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

12.5.12 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC0) is set to 1 for extension code reception and an interrupt request (INTIIC0) is issued at the falling edge of the eighth clock. The local address stored in slave address register 0 (SVA0) is not affected.

(2) If “11110××0” is set to SVA0 by a 10-bit address transfer and “11110××0” is transferred from the master device, the results are as follows. Note that INTIIC0 occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC0 = 1
- Seven bits of data match: COI0 = 1

Remark EXC0: Bit 5 of IIC status register 0 (IICS0)
COI0: Bit 4 of IIC status register 0 (IICS0)

(3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
If the extension code is received while a slave device is operating, then the slave device is participating in communication even if its address does not match.
For example, after the extension code is received, if you do not wish to operate the target device as a slave device, set bit 6 (LREL0) of the IIC control register 0 (IICC0) to 1 to set the standby mode for the next communication operation.

Table 12-5. Extension Code Bit Definitions

Slave Address	R/W Bit	Description
0 0 0 0 0 0 0	0	General call address
0 0 0 0 0 0 0	1	Start byte
0 0 0 0 0 0 1	×	C-BUS address
0 0 0 0 0 1 0	×	Address that is reserved for different bus format
1 1 1 1 0 x x	×	10-bit slave address specification

12.5.13 Arbitration

When several master devices simultaneously generate a start condition (when STT0 is set to 1 before STD0 is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (ALD0) in IIC status register 0 (IICS0) is set (1) via the timing by which the arbitration loss occurred, and the SCL0 and SDA0 lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 = 1 setting that has been made by software.

For details of interrupt request timing, see **12.5.9 Interrupt request (INTIIC0) generation timing and wait control**.

Remark STD0: Bit 1 of IIC status register 0 (IICS0)
STT0: Bit 1 of IIC control register 0 (IICC0)

Figure 12-20. Arbitration Timing Example

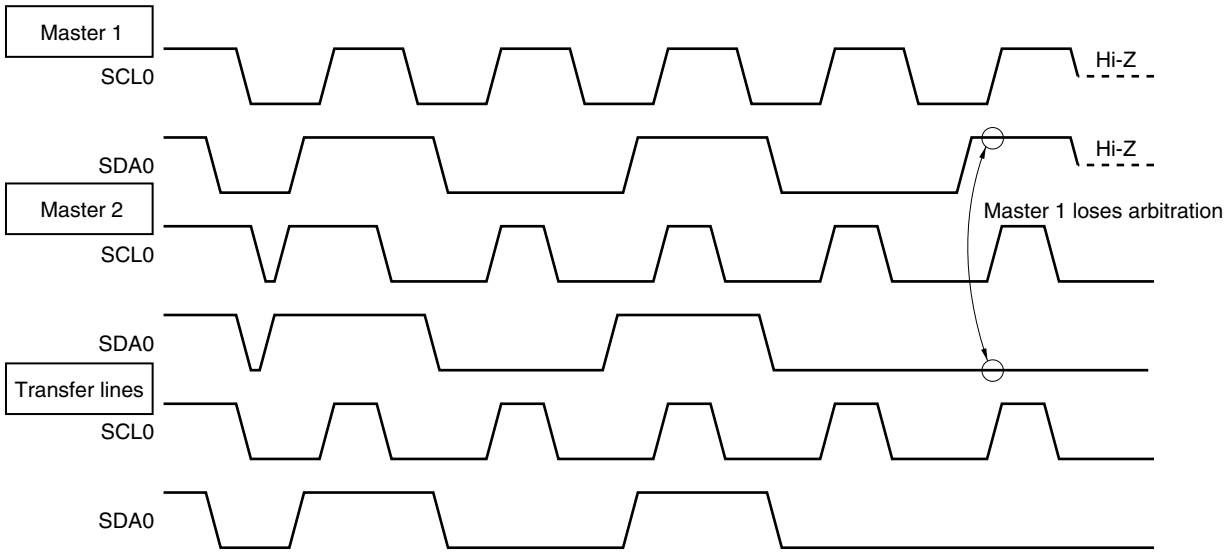


Table 12-6. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[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 SPIE0 = 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 SPIE0 = 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 WTIM0 (bit 3 of IIC control register 0 (IICC0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0 = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a chance that arbitration will occur, set SPIE0 = 1 for master device operation.

Remark SPIE0: Bit 4 of IIC control register 0 (IICC0)

12.5.14 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIIC0) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIIC0 signal from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

However, when a stop condition is detected, bit 4 (SPIE0) of IIC control register 0 (IICC0) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

12.5.15 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IIC flag register 0 (IICF0) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. 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 when bit 6 (LREL0) of IIC control register 0 (IICC0) was set to 1).

If bit 1 (STT0) of IICC0 is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.

If an address is written to IIC shift register 0 (IIC0) after bit 4 (SPIE0) of IICC0 was set to 1, and it was detected by generation of an interrupt request signal (INTIIC0) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to IIC0 before the stop condition is detected is invalid.

When STT0 has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released ..... a start condition is generated
- If the bus has not been released (standby mode)..... communication reservation

Check whether the communication reservation operates or not by using MSTS0 (bit 7 of IIC status register 0 (IICS0)) after STT0 is set to 1 and the wait time elapses.

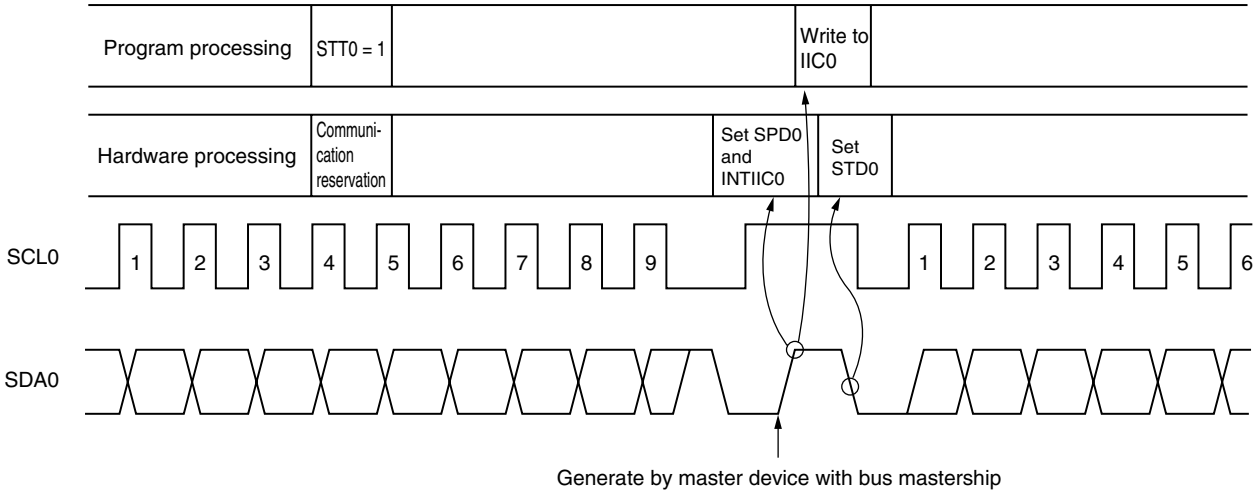
The wait periods, which should be set via software, are listed in Table 12-6.

Table 12-7. Wait Periods

CLX0	SMC0	CL01	CL00	Wait Period
0	0	0	0	43 clocks
0	0	0	1	85 clocks
0	0	1	0	101 clocks
0	0	1	1	23 clocks
0	1	0	0	27 clocks
0	1	0	1	
0	1	1	0	51 clocks
0	1	1	1	15 clocks
1	1	0	0	
1	1	0	1	
1	1	1	0	27 clocks
1	1	1	1	9 clocks

Figure 12-21 shows the communication reservation timing.

Figure 12-21. Communication Reservation Timing



Remark IIC0: IIC shift register 0
STT0: Bit 1 of IIC control register 0 (IICC0)
STD0: Bit 1 of IIC status register 0 (IICS0)
SPD0: Bit 0 of IIC status register 0 (IICS0)

Communication reservations are accepted via the following timing. After bit 1 (STD0) of IIC status register 0 (IICS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IIC control register 0 (IICC0) to 1 before a stop condition is detected.

Figure 12-22. Timing for Accepting Communication Reservations

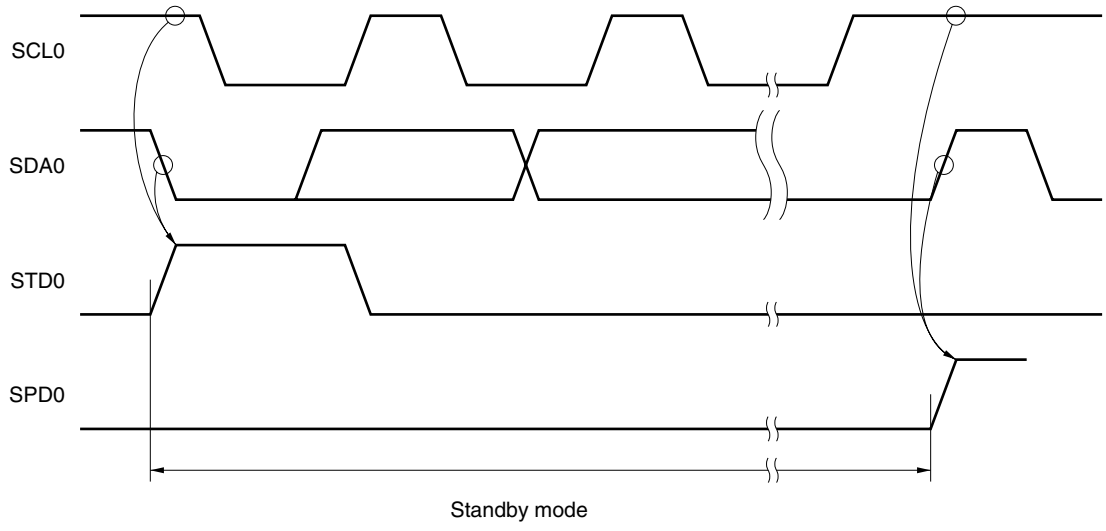
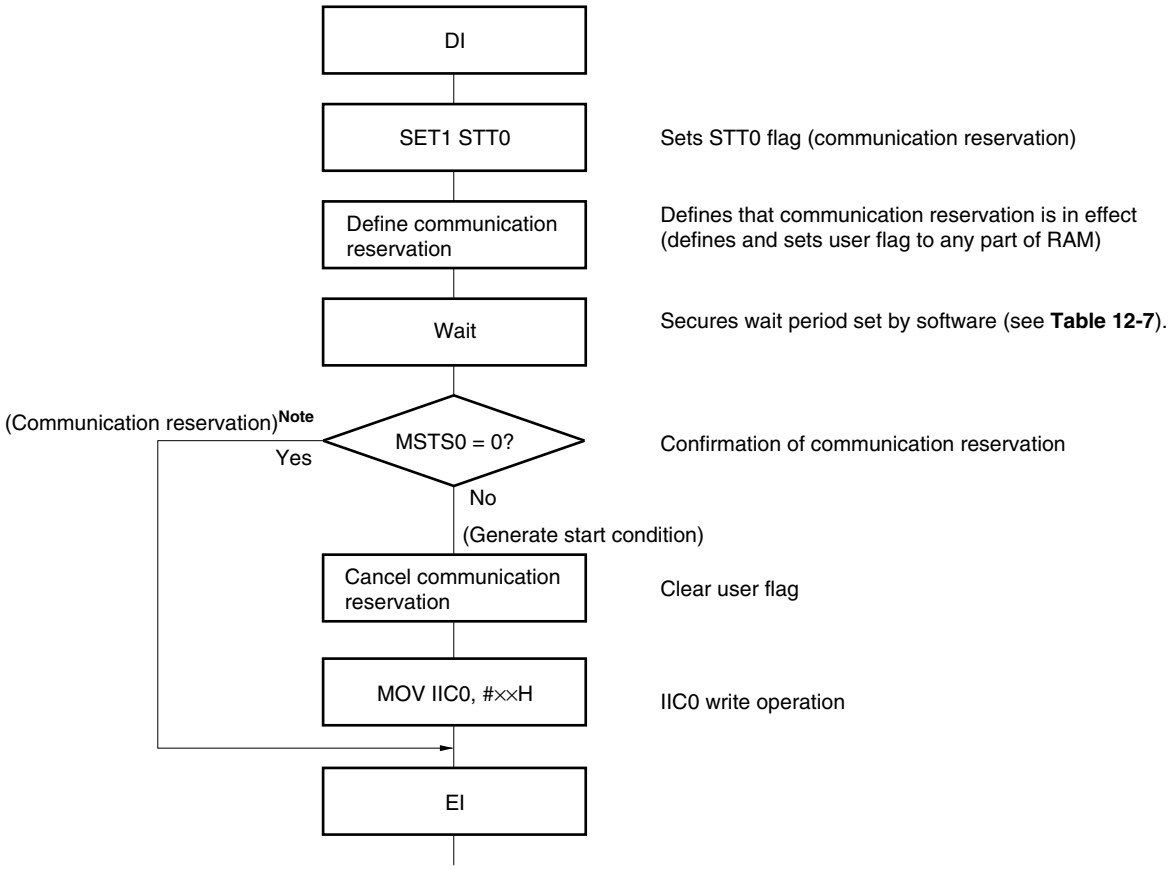


Figure 12-23 shows the communication reservation protocol.

Figure 12-23. Communication Reservation Protocol



Note The communication reservation operation executes a write to IIC shift register 0 (IIC0) when a stop condition interrupt request occurs.

Remark STT0: Bit 1 of IIC control register 0 (IICC0)
MSTS0: Bit 7 of IIC status register 0 (IICS0)
IIC0: IIC shift register 0

(2) When communication reservation function is disabled (bit 0 (IICRSV) of IIC flag register 0 (IICF0) = 1)

When bit 1 (STT0) of IIC control register 0 (IICC0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released when bit 6 (LREL0) of IICC0 was set to 1)

To confirm whether the start condition was generated or request was rejected, check STCF (bit 7 of IICF0). It takes up to 5 clocks until STCF is set to 1 after setting STT0 = 1. Therefore, secure the time by software.

12.5.16 Cautions

(1) When STCEN (bit 1 of IIC flag register 0 (IICF0)) = 0

Immediately after I^2C operation is enabled (IICE0 = 1), the bus communication status (IICBSY (bit 6 of IICF0) = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

<1> Set IIC clock select register 0 (IICCL0).
<2> Set bit 7 (IICE0) of IIC control register 0 (IICC0) to 1.
<3> Set bit 0 (SPT0) of IICC0 to 1.

(2) When STCEN = 1

Immediately after I^2C operation is enabled (IICE0 = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT0 (bit 1 of IIC control register 0 (IICC0)) = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I^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 (SPIE0) of IICC0 to 0 to disable generation of an interrupt request signal (INTIIC0) when the stop condition is detected.
<2> Set bit 7 (IICE0) of IICC0 to 1 to enable the operation of I^2C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL0) of IICC0 to 1 before $\overline{ACK}$ is returned (4 to 80 clocks after setting IICE0 to 1), to forcibly disable detection.

(4) Determine the transfer clock frequency by using SMC0, CL01, CL00 (bits 3, 1, and 0 of IICL0), and CLX0 (bit 0 of IICX0) before enabling the operation (IICE0 = 1). To change the transfer clock frequency, clear IICE0 to 0 once.

(5) Setting STT0 and SPT0 (bits 1 and 0 of IICC0) again after they are set and before they are cleared to 0 is prohibited.

(6) When transmission is reserved, set SPIE0 (bit 4 of IICL0) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to IIC0 after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set SPIE0 to 1 when MSTS0 (bit 7 of IICS0) is detected by software.

12.5.17 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the 78K0R/KE3 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/KE3 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/KE3 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

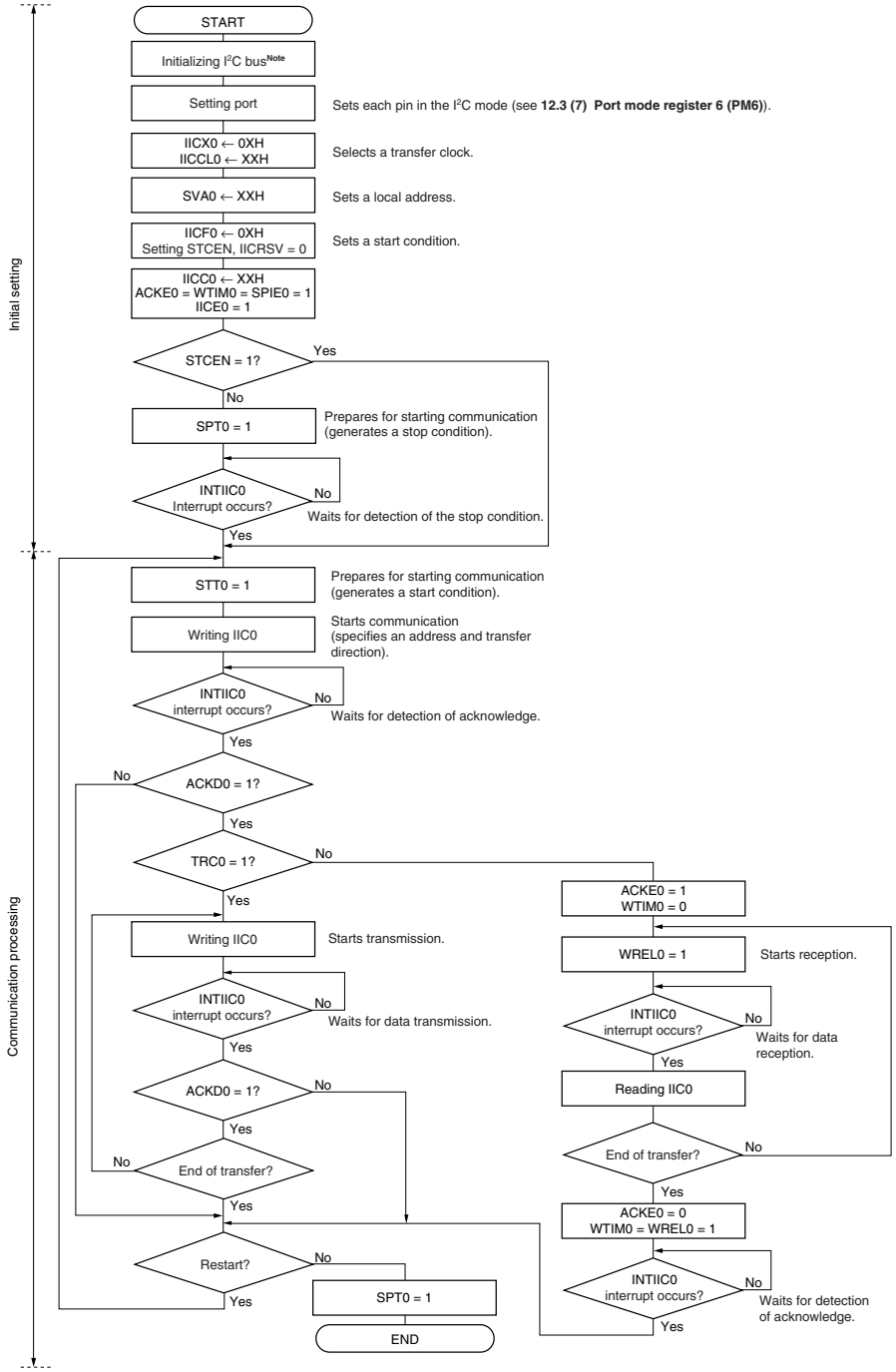
An example of when the 78K0R/KE3 is used as the I^2C bus slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIIC0 interrupt occurrence (communication waiting). When an INTIIC0 interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

(1) Master operation in single-master system

Figure 12-24. Master Operation in Single-Master System

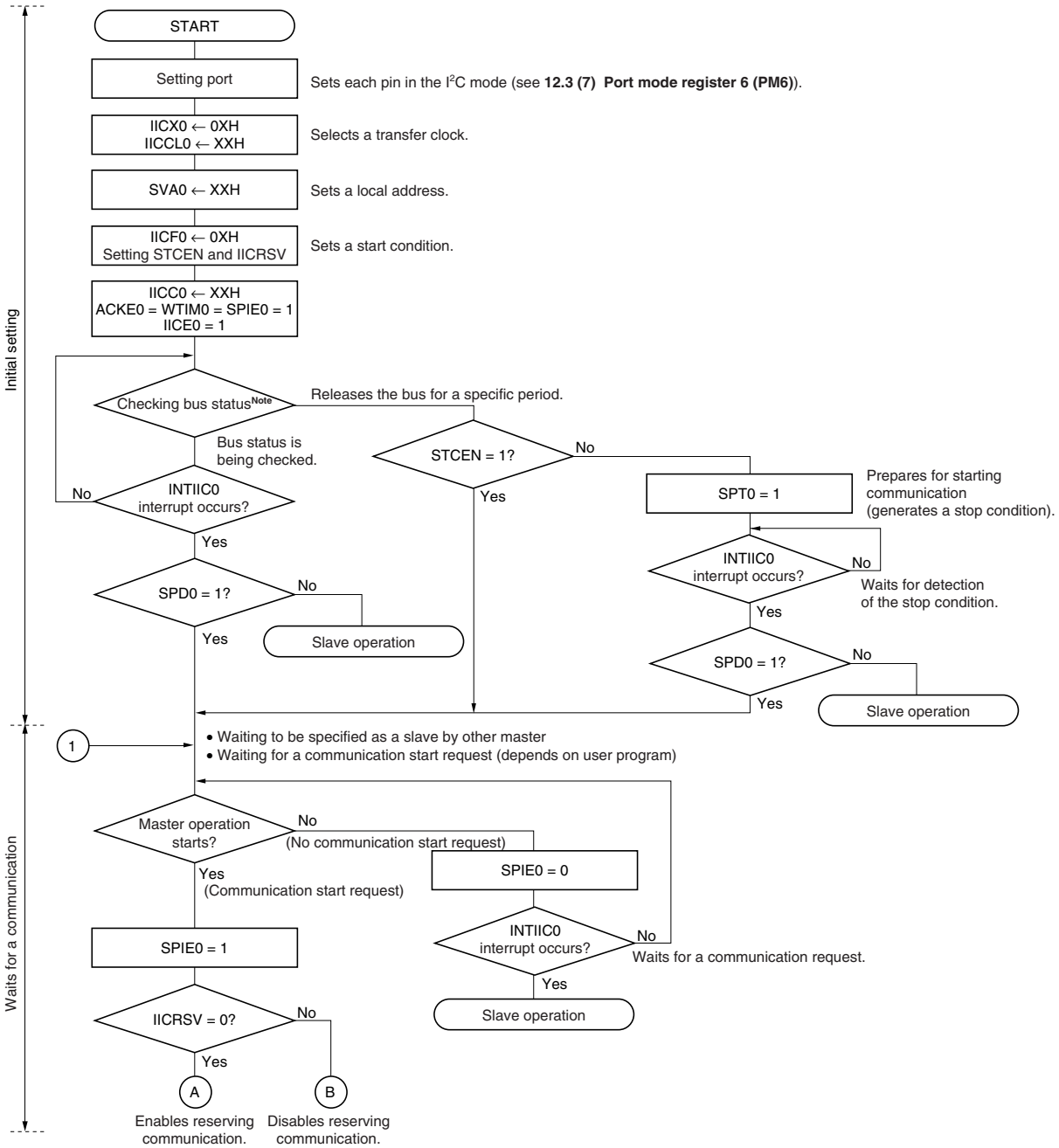


Note Release (SCL0 and SDA0 pins = high level) the I^2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDA0 pin, for example, set the SCL0 pin in the output port mode, and output a clock pulse from the output port until the 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 12-25. Master Operation in Multi-Master System (1/3)



Note Confirm that the bus is released (CLD0 bit = 1, DAD0 bit = 1) for a specific period (for example, for a period of one frame). If the SDA0 pin is constantly at low level, decide whether to release the I²C bus (SCL0 and SDA0 pins = high level) in conformance with the specifications of the product that is communicating.

Figure 12-25. Master Operation in Multi-Master System (2/3)

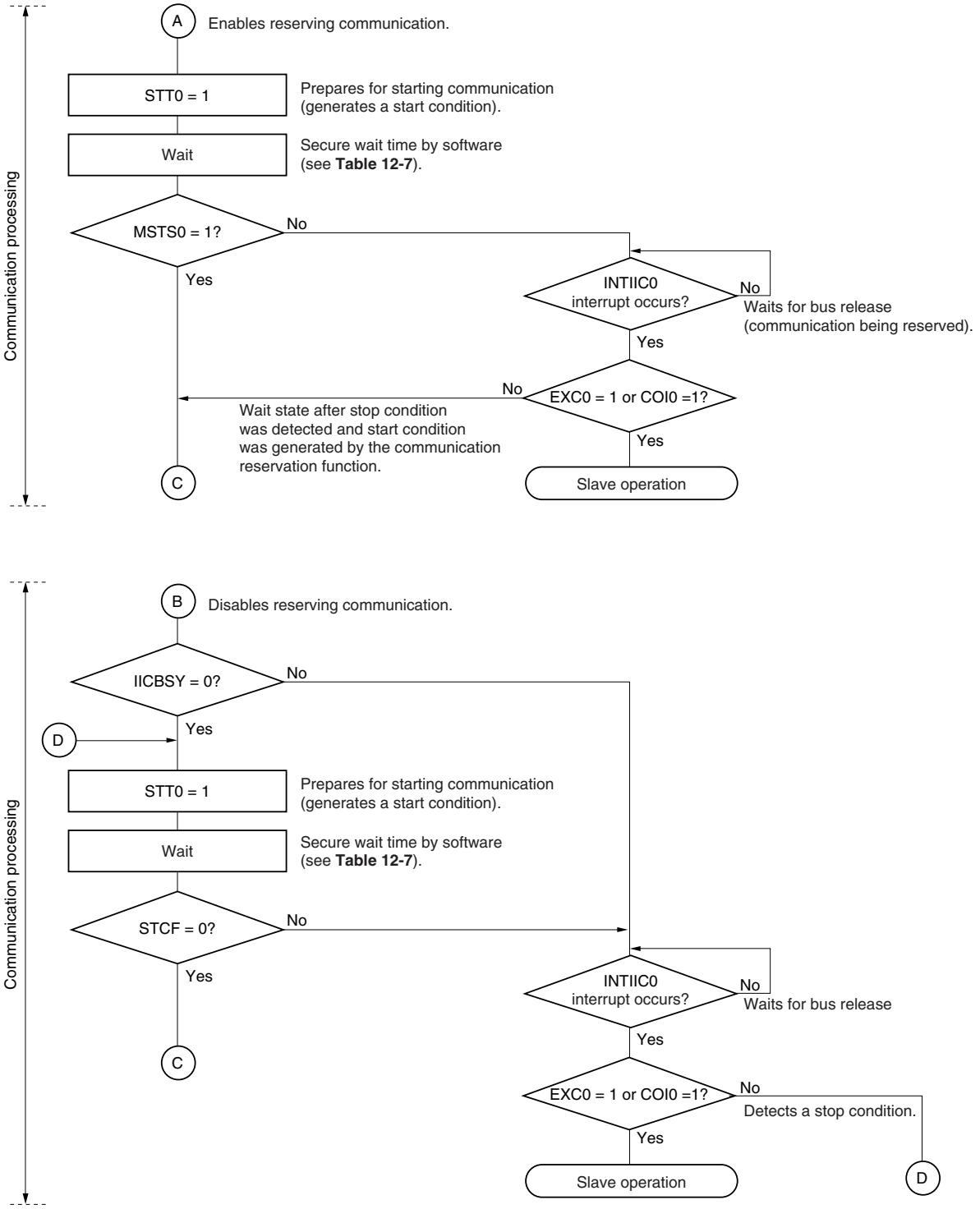
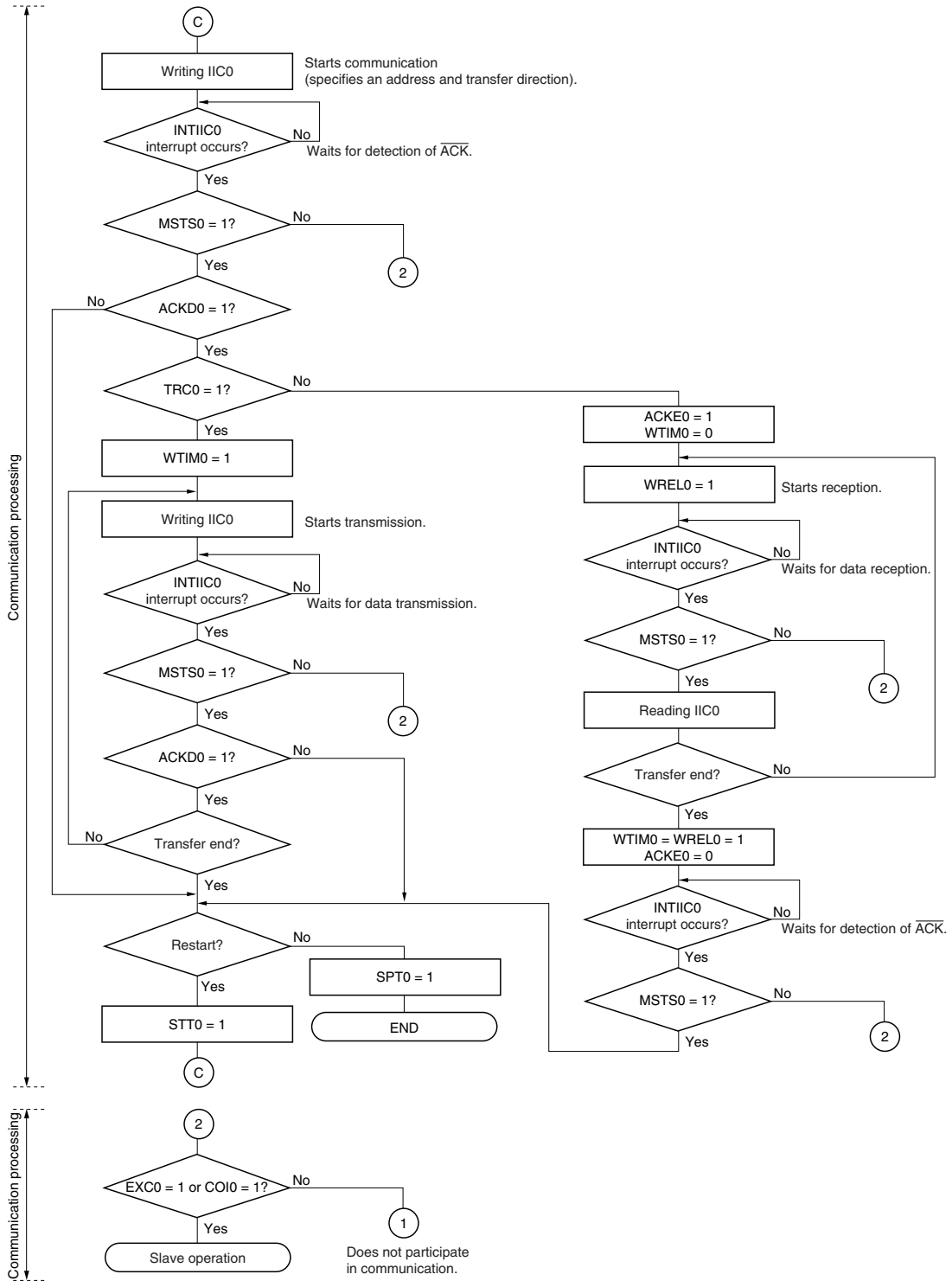


Figure 12-25. Master Operation in Multi-Master System (3/3)



Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. To use the device as a master in a multi-master system, read the MSTS0 bit each time interrupt INTIIC0 has occurred to check the arbitration result.

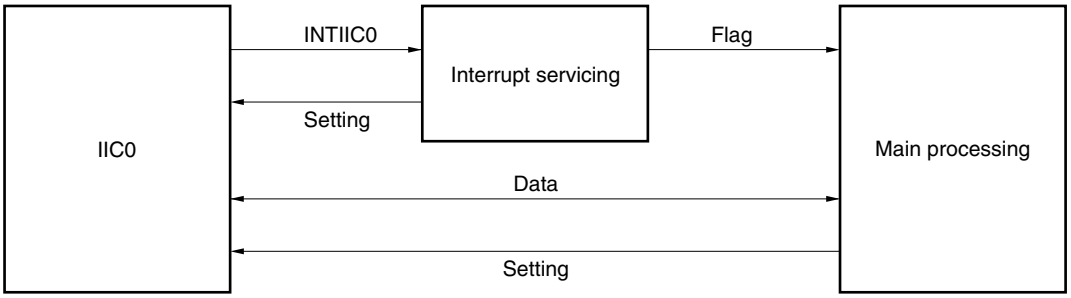
3. To use the device as a slave in a multi-master system, check the status by using the IICS0 and IICF0 registers each time interrupt INTIIC0 has occurred, and determine the processing to be performed next.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIIC0 interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIIC0 interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIIC0.

<1> Communication mode flag

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of ACK from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIIC0 interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as TRC0.

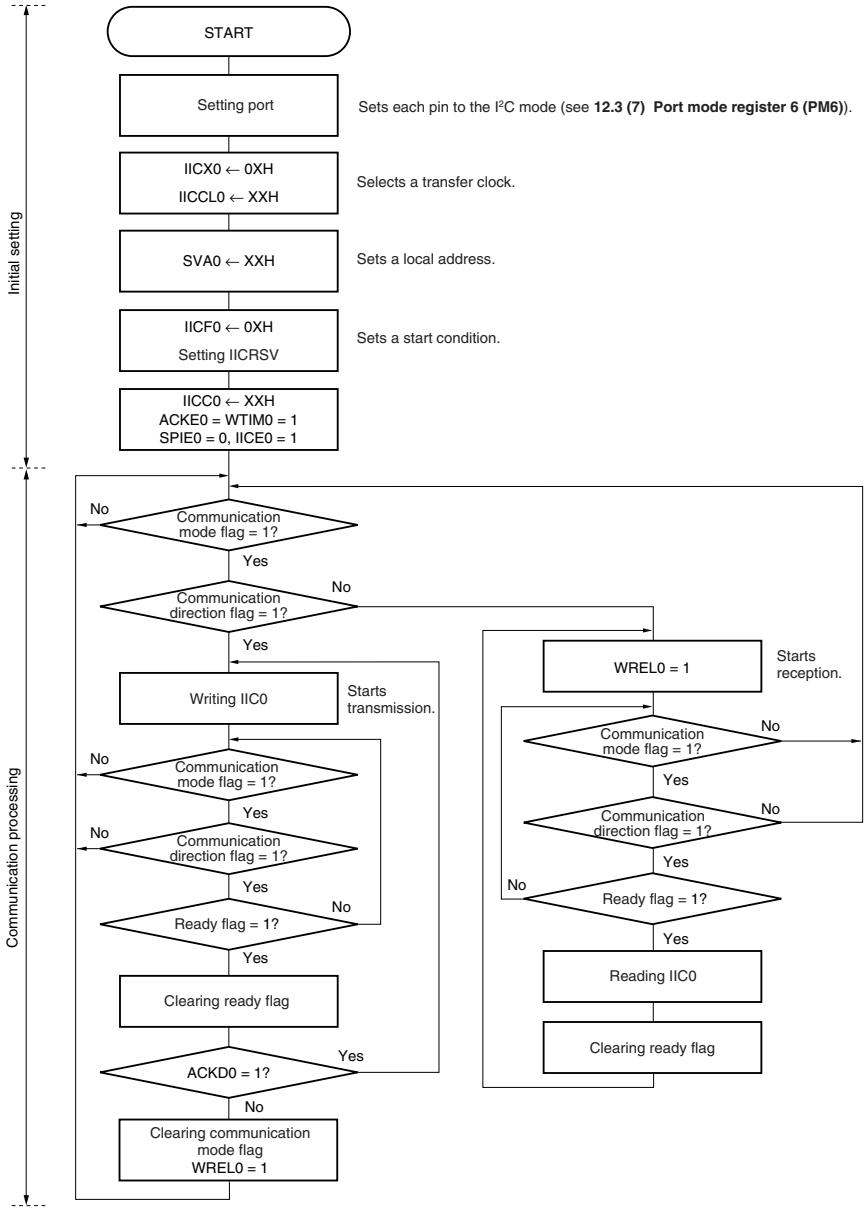
The main processing of the slave operation is explained next.

Start serial interface IIC0 and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns $\overline{\text{ACK}}$. If ACK is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, ACK is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Figure 12-26. Slave Operation Flowchart (1)



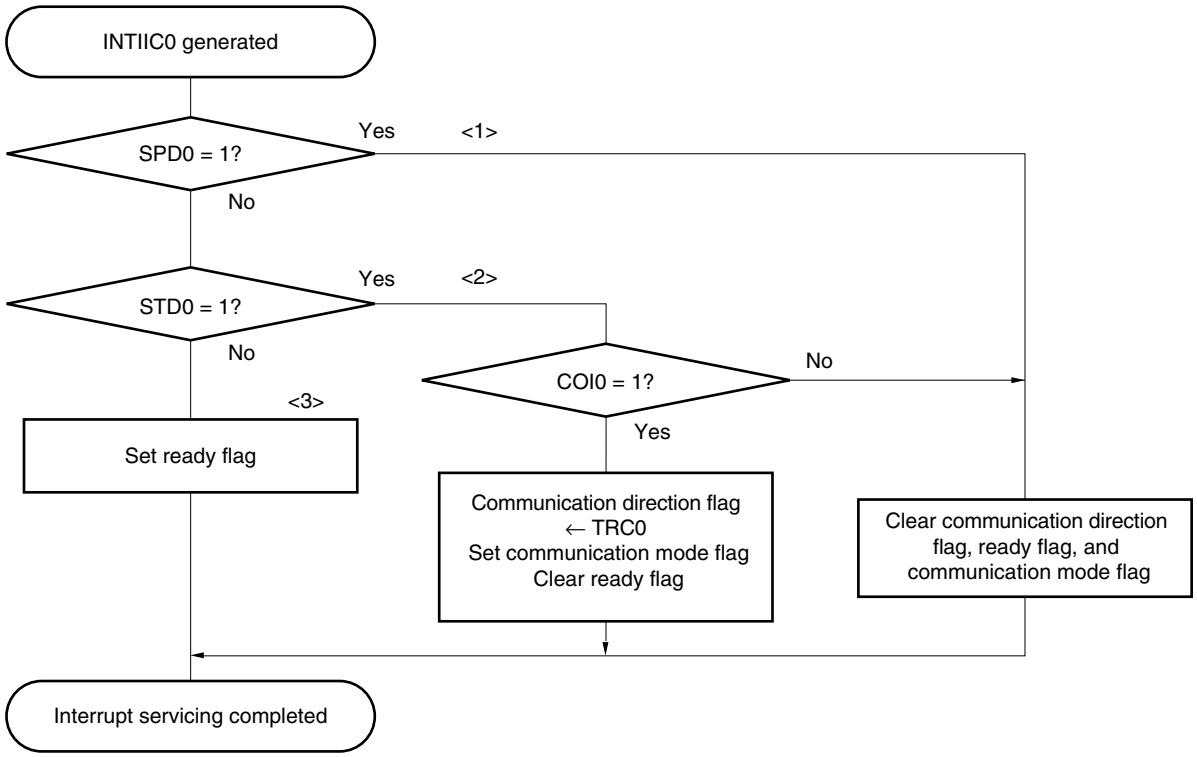
Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

An example of the processing procedure of the slave with the INTIIC0 interrupt is explained below (processing is performed assuming that no extension code is used). The INTIIC0 interrupt checks the status, and the following operations are performed.

<1> Communication is stopped if the stop condition is issued.
<2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).
<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in **Figure 12-27 Slave Operation Flowchart (2)**.

Figure 12-27. Slave Operation Flowchart (2)



12.5.18 Timing of I²C interrupt request (INTIIC0) occurrence

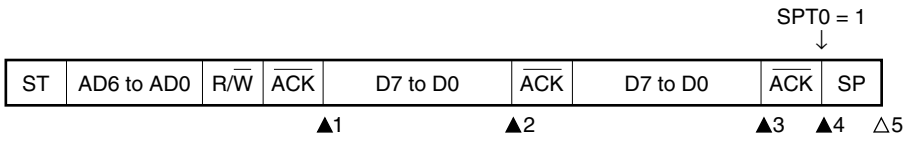
The timing of transmitting or receiving data and generation of interrupt request signal INTIIC0, and the value of the IICS0 register when the INTIIC0 signal is generated are shown below.

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 WTIM0 = 0

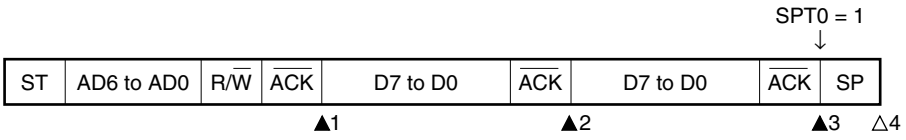


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B
▲3: IICS0 = 1000×000B (Sets WTIM0 to 1)[Note]
▲4: IICS0 = 1000××00B (Sets SPT0 to 1)[Note]
△5: IICS0 = 00000001B

Note To generate a stop condition, set WTIM0 to 1 and change the timing for generating the INTIIC0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

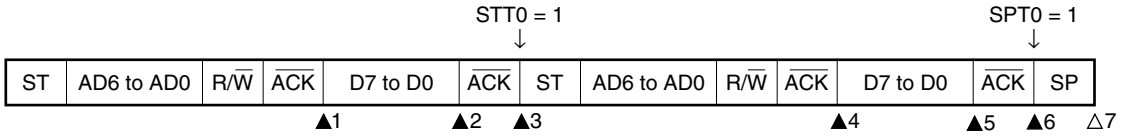


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B
▲3: IICS0 = 1000××00B (Sets SPT0 to 1)
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets WTIM0 to 1)[Note 1]
▲3: IICS0 = 1000××00B (Clears WTIM0 to 0[Note 2], sets STT0 to 1)
▲4: IICS0 = 1000×110B
▲5: IICS0 = 1000×000B (Sets WTIM0 to 1)[Note 3]
▲6: IICS0 = 1000××00B (Sets SPT0 to 1)
△7: IICS0 = 00000001B

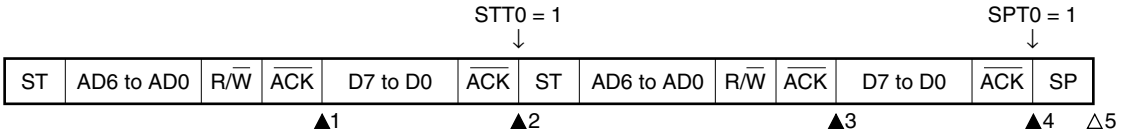
Notes 1. To generate a start condition, set WTIM0 to 1 and change the timing for generating the INTIIC0 interrupt request signal.

2. Clear WTIM0 to 0 to restore the original setting.

3. To generate a stop condition, set WTIM0 to 1 and change the timing for generating the INTIIC0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

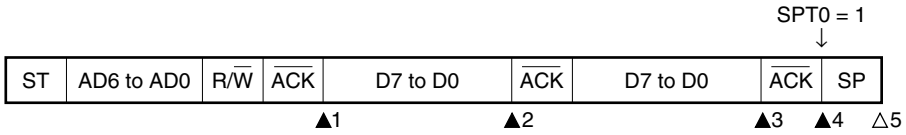


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets STT0 to 1)
▲3: IICS0 = 1000×110B
▲4: IICS0 = 1000××00B (Sets SPT0 to 1)
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0

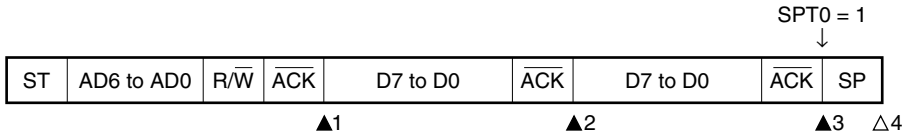


▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×000B
▲3: IICS0 = 1010×000B (Sets WTIM0 to 1)[Note]
▲4: IICS0 = 1010××00B (Sets SPT0 to 1)
△5: IICS0 = 00000001B

Note To generate a stop condition, set WTIM0 to 1 and change the timing for generating the INTIIC0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



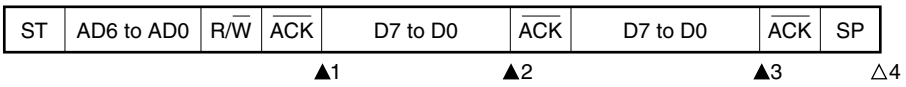
▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×100B
▲3: IICS0 = 1010××00B (Sets SPT0 to 1)
△4: IICS0 = 00001001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

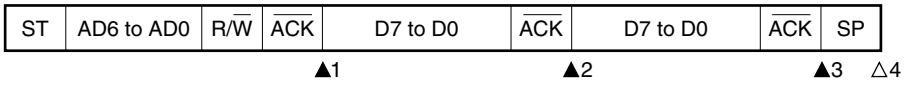
(i) When WTIM0 = 0



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

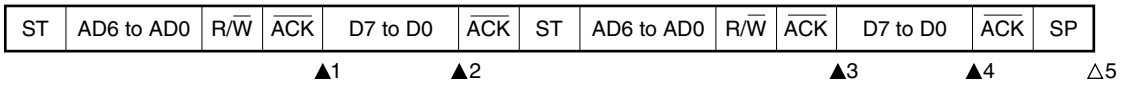


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

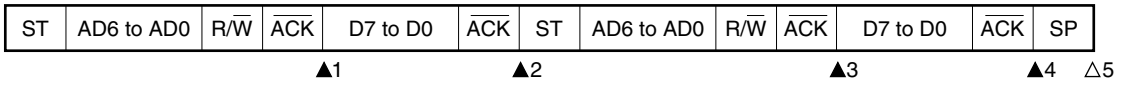
(i) When WTIM0 = 0 (after restart, matches with SVA0)



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)

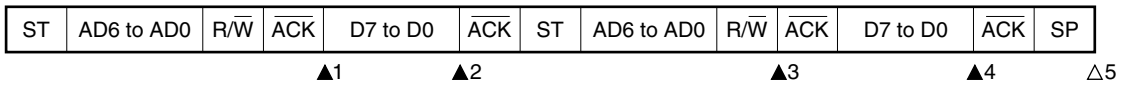


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

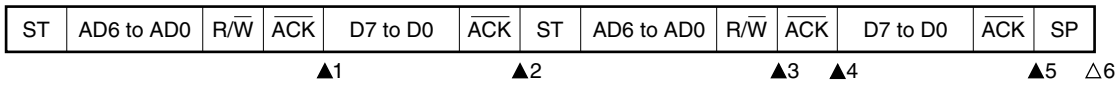
(i) When WTIM0 = 0 (after restart, does not match address (= extension code))



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= extension code))

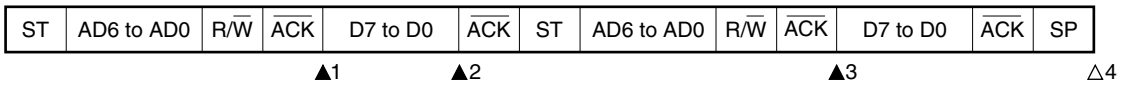


▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×110B
▲5: IICS0 = 0010××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

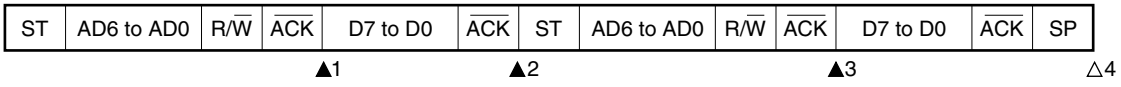
(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 00000110B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))



▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 00000110B
△4: IICS0 = 00000001B

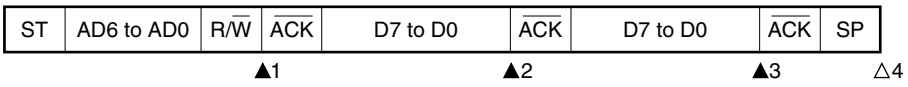
Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(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

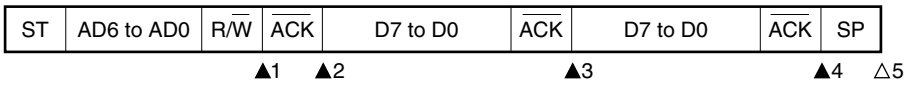
(i) When WTIM0 = 0



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

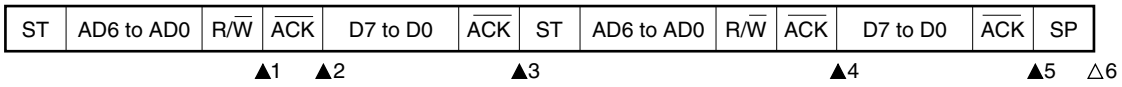
(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, matches SVA0)

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches SVA0)

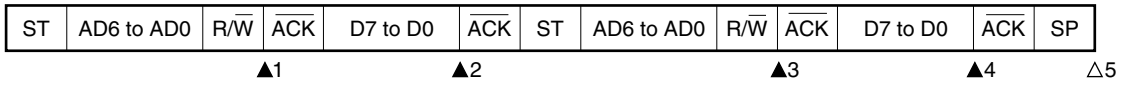


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0001×110B
▲5: IICS0 = 0001××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

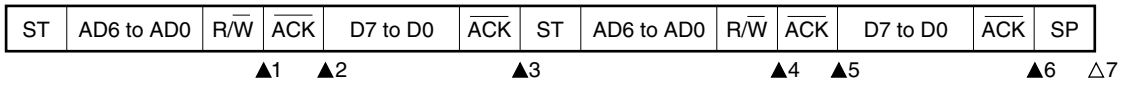
(i) When WTIM0 = 0 (after restart, extension code reception)



▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)

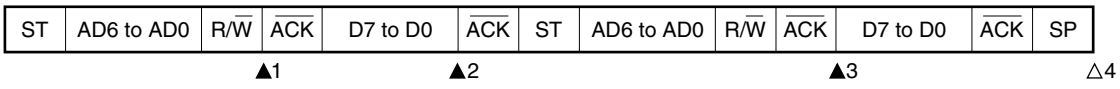


▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0010×010B
▲5: IICS0 = 0010×110B
▲6: IICS0 = 0010××00B
△7: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

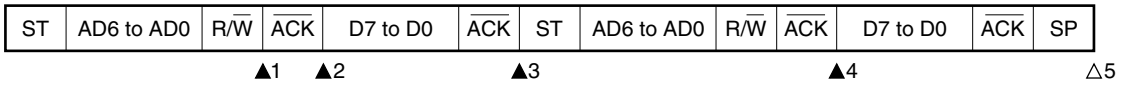
(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))



▲1: IICS0 = 00100010B
▲2: IICS0 = 00100000B
▲3: IICS0 = 00000110B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))



▲1: IICS0 = 00100010B
▲2: IICS0 = 00100110B
▲3: IICS0 = 00100×00B
▲4: IICS0 = 00000110B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

△1

△1: IICS0 = 00000001B

Remark △: Generated only when SPIE0 = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS0 bit each time interrupt request signal INTIIC0 has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

(i) When WTIM0 = 0

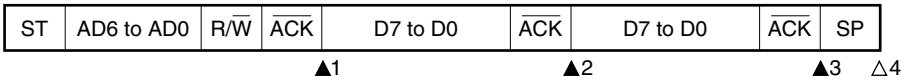
ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 △4

▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

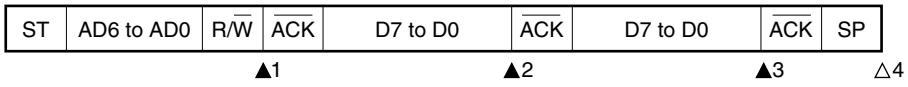


▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

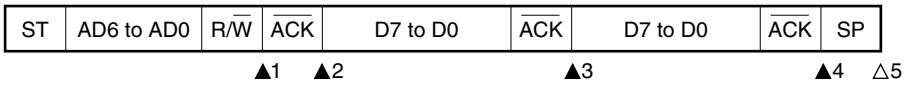
(i) When WTIM0 = 0



▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



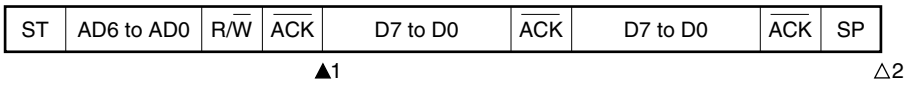
▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 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 MSTS0 bit each time interrupt request signal INTIIC0 has occurred to check the arbitration result.

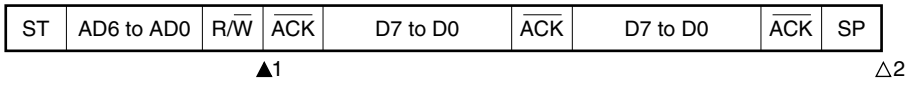
(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)



▲1: IICS0 = 01000110B
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(b) When arbitration loss occurs during transmission of extension code

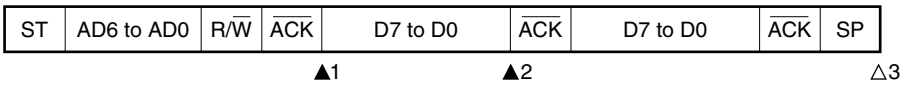


▲1: IICS0 = 0110×010B
Sets LREL0 = 1 by software
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

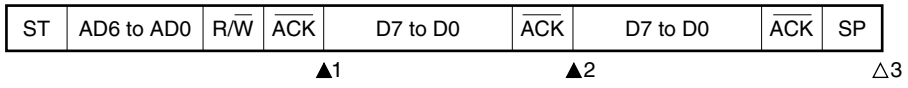
(i) When WTIM0 = 0



▲1: IICS0 = 10001110B
▲2: IICS0 = 01000000B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1

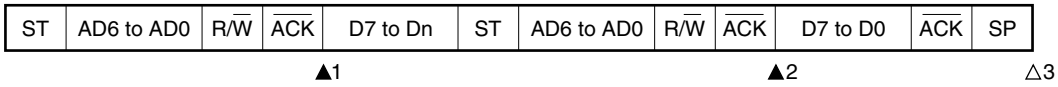


▲1: IICS0 = 10001110B
▲2: IICS0 = 01000100B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

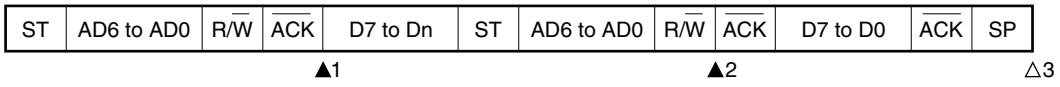
(i) Not extension code (Example: unmatches with SVA0)



▲1: IICS0 = 1000×110B
▲2: IICS0 = 01000110B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

(ii) Extension code



▲1: IICS0 = 1000×110B

▲2: IICS0 = 01100010B

Sets LREL0 = 1 by software

△3: IICS0 = 00000001B

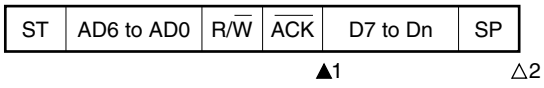
Remark ▲: Always generated

△: Generated only when SPIE0 = 1

×: Don't care

n = 6 to 0

(e) When loss occurs due to stop condition during data transfer



▲1: IICS0 = 10000110B

△2: IICS0 = 01000001B

Remark ▲: Always generated

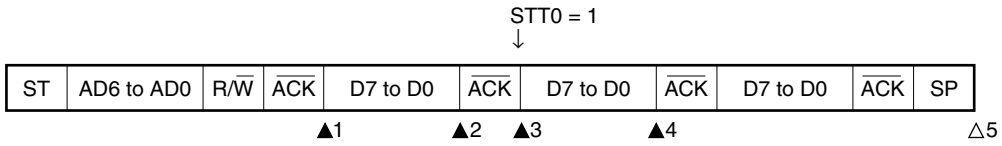
△: Generated only when SPIE0 = 1

×: Don't care

n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

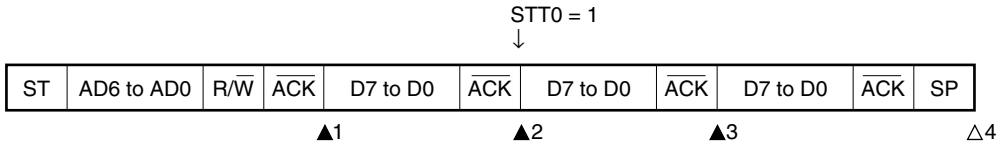
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets WTIM0 to 1)
▲3: IICS0 = 1000×100B (Clears WTIM0 to 0)
▲4: IICS0 = 01000000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

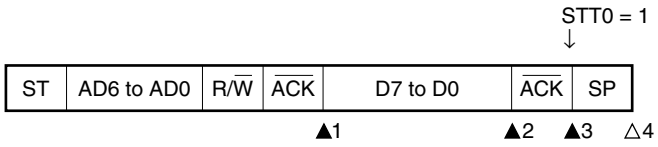


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets STT0 to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

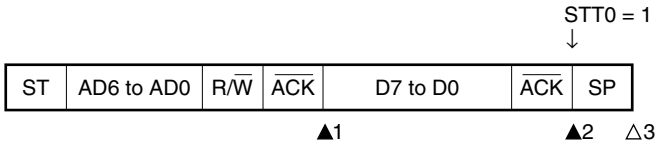
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets WTIM0 to 1)
▲3: IICS0 = 1000××00B (Sets STT0 to 1)
△4: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

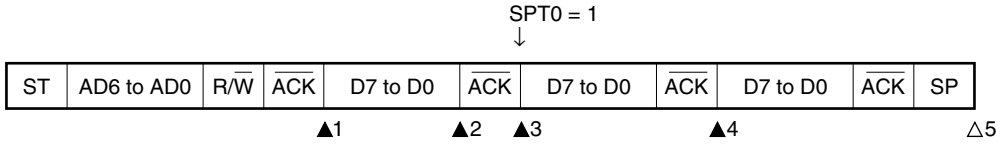


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets STT0 to 1)
△3: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

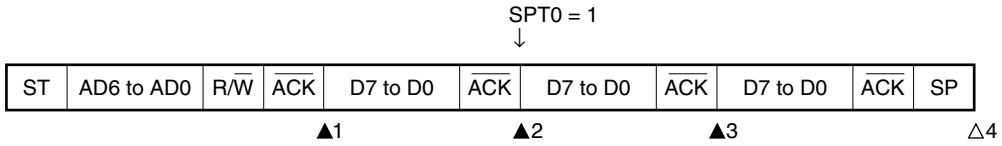
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets WTIM0 to 1)
▲3: IICS0 = 1000×100B (Clears WTIM0 to 0)
▲4: IICS0 = 01000100B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets SPT0 to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 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 TRC0 bit (bit 3 of IIC status register 0 (IICS0)), which specifies the data transfer direction, and then starts serial communication with the slave device.

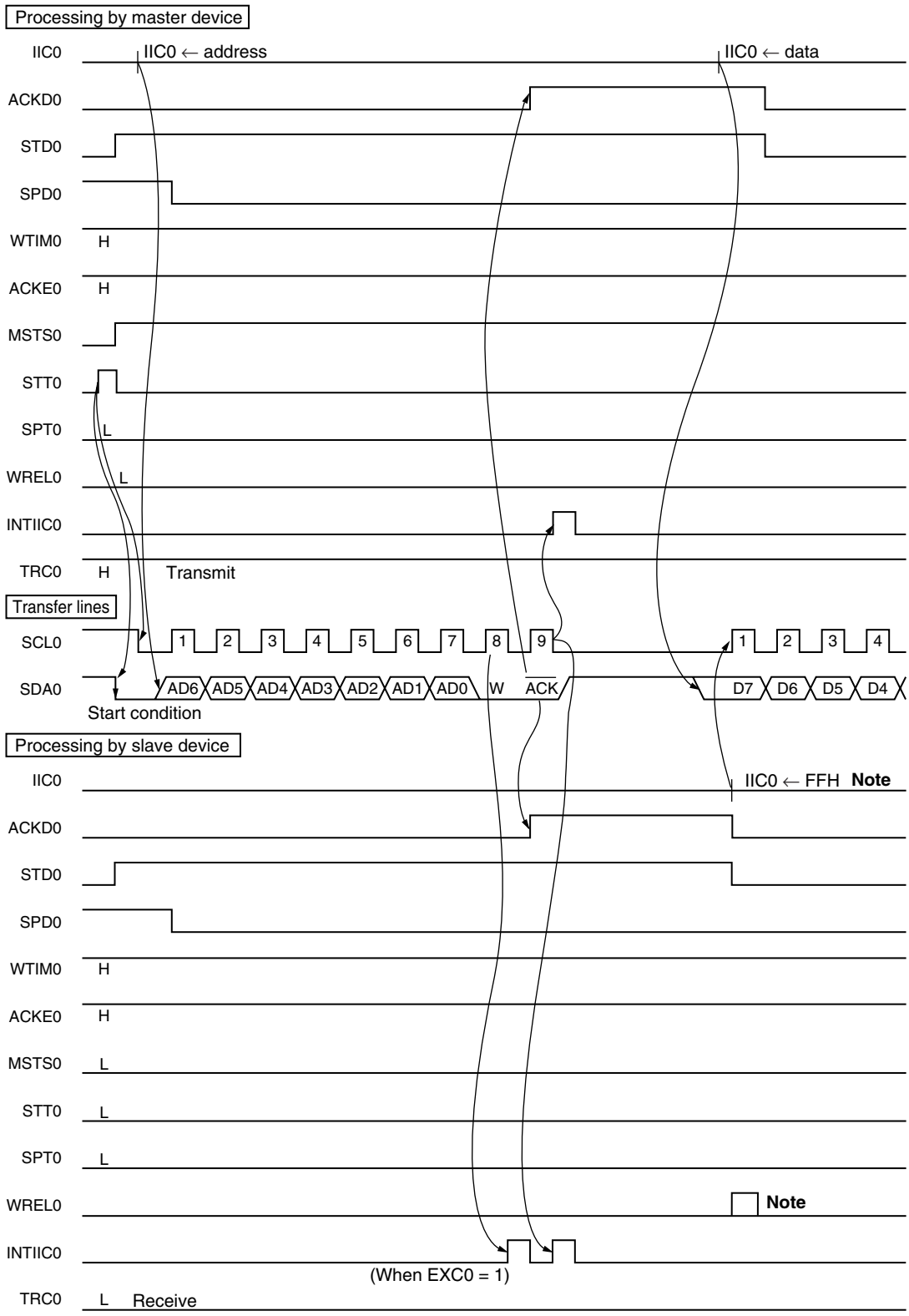
Figures 12-28 and 12-29 show timing charts of the data communication.

IIC shift register 0 (IIC0)'s shift operation is synchronized with the falling edge of the serial clock (SCL0). The transmit data is transferred to the SO0 latch and is output (MSB first) via the SDA0 pin.

Data input via the SDA0 pin is captured into IIC0 at the rising edge of SCL0.

Figure 12-28. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

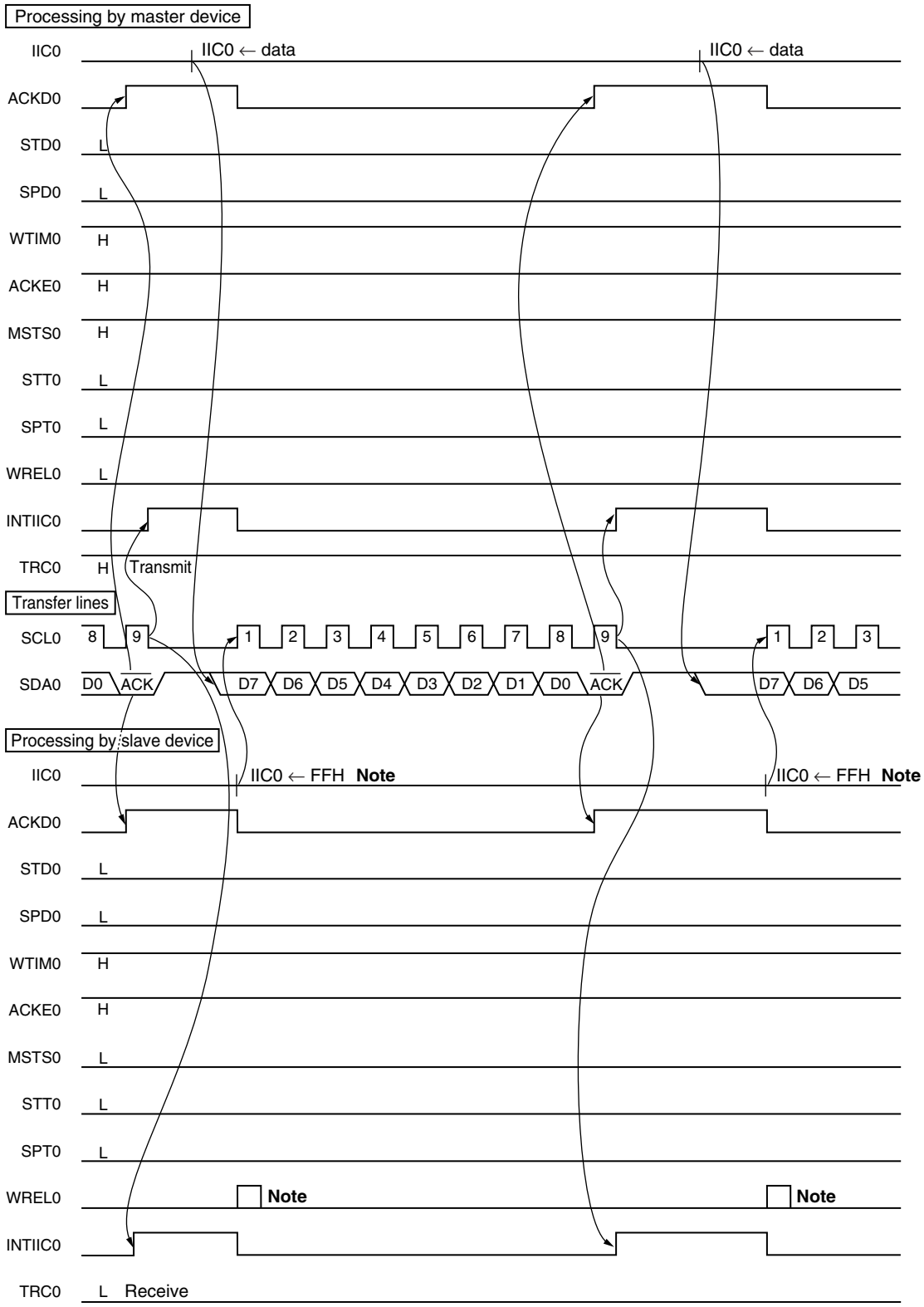
(1) Start condition ~ address



Note To cancel slave wait, write “FFH” to IIC0 or set WREL0.

Figure 12-28. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

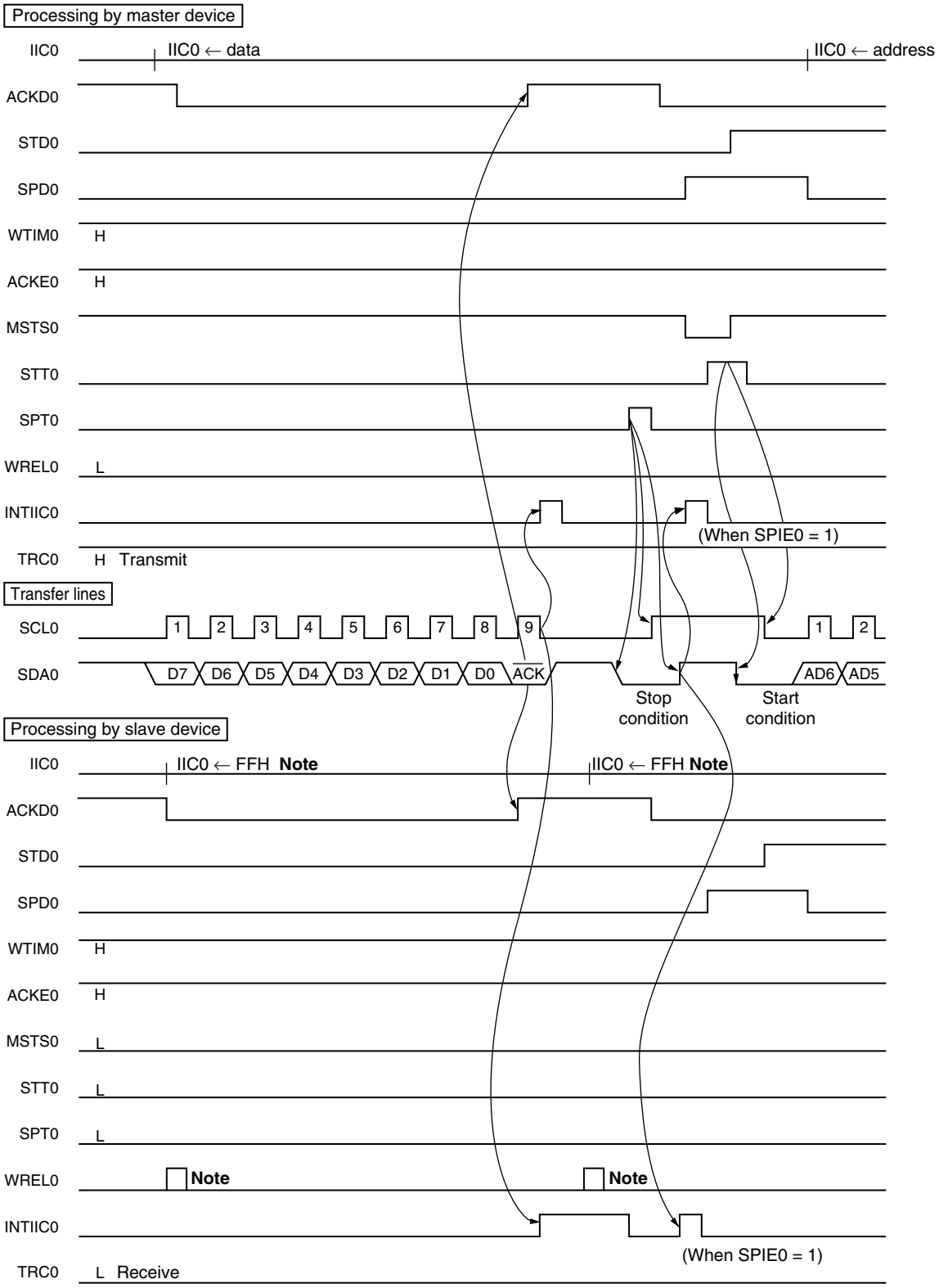
(2) Data



Note To cancel slave wait, write "FFH" to IIC0 or set WREL0.

Figure 12-28. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

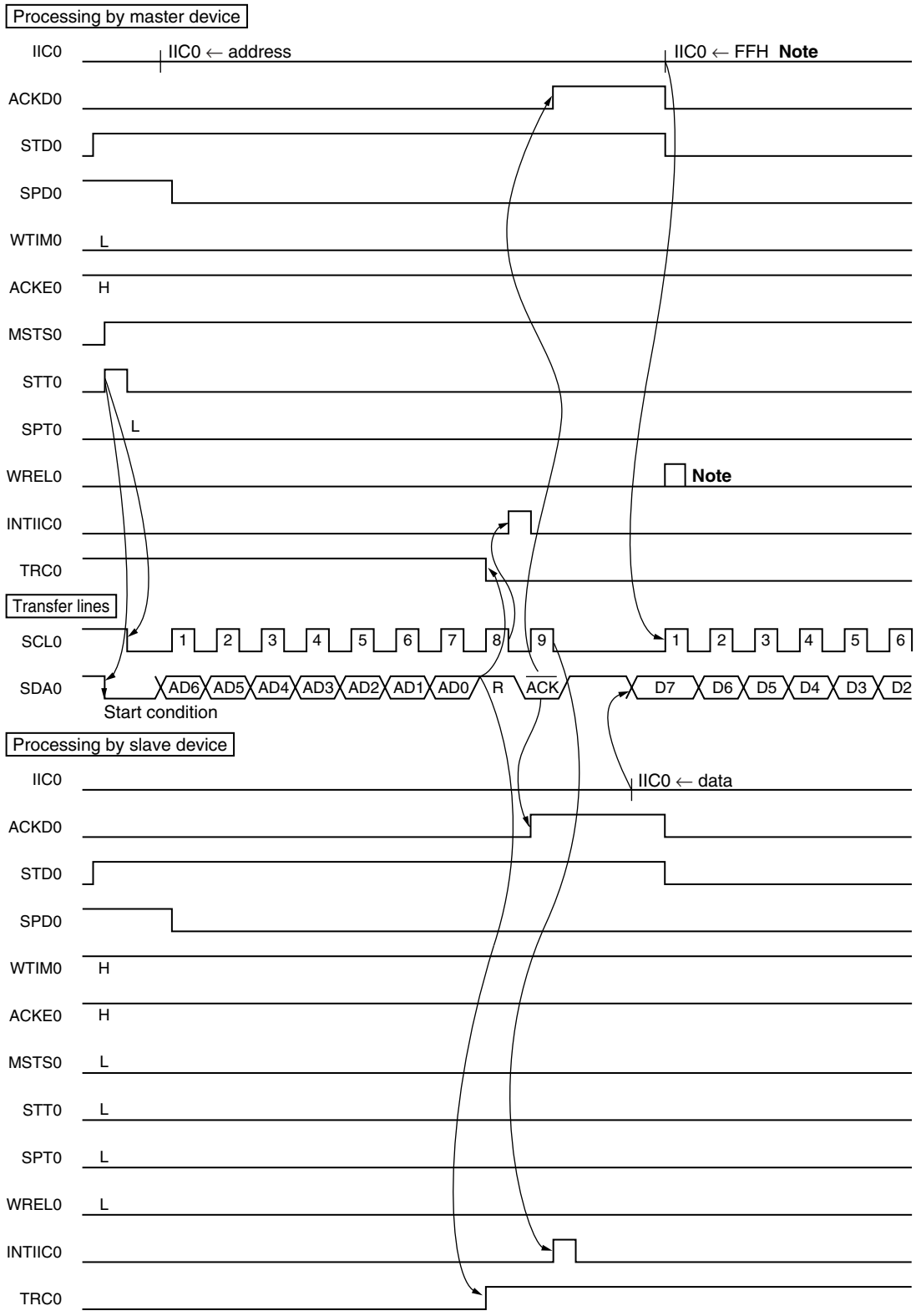
(3) Stop condition



Note To cancel slave wait, write “FFH” to IIC0 or set WREL0.

Figure 12-29. Example of Slave to Master Communication
(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)

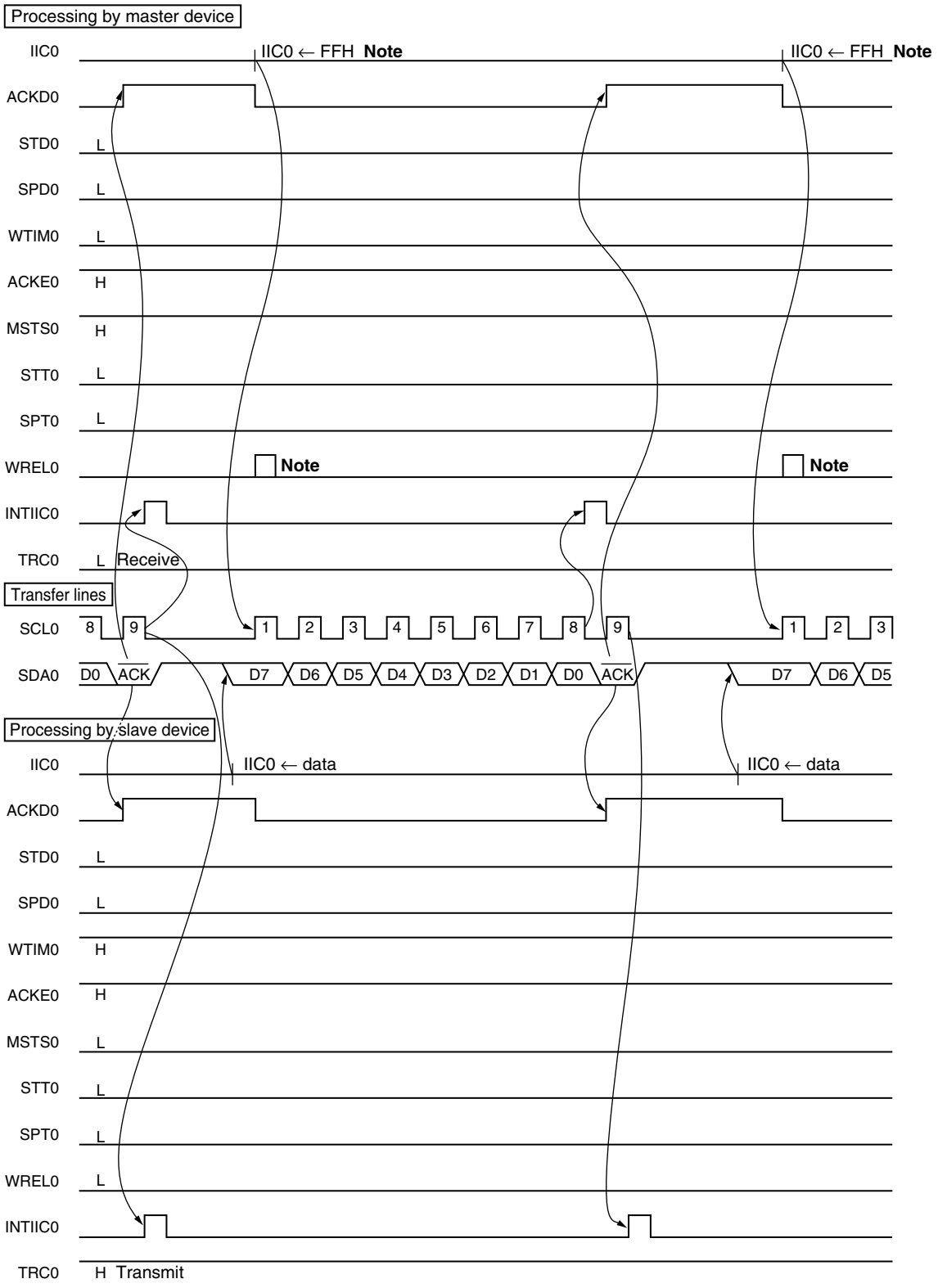
(1) Start condition ~ address



Note To cancel master wait, write “FFH” to IIC0 or set WREL0.

Figure 12-29. Example of Slave to Master Communication
(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)

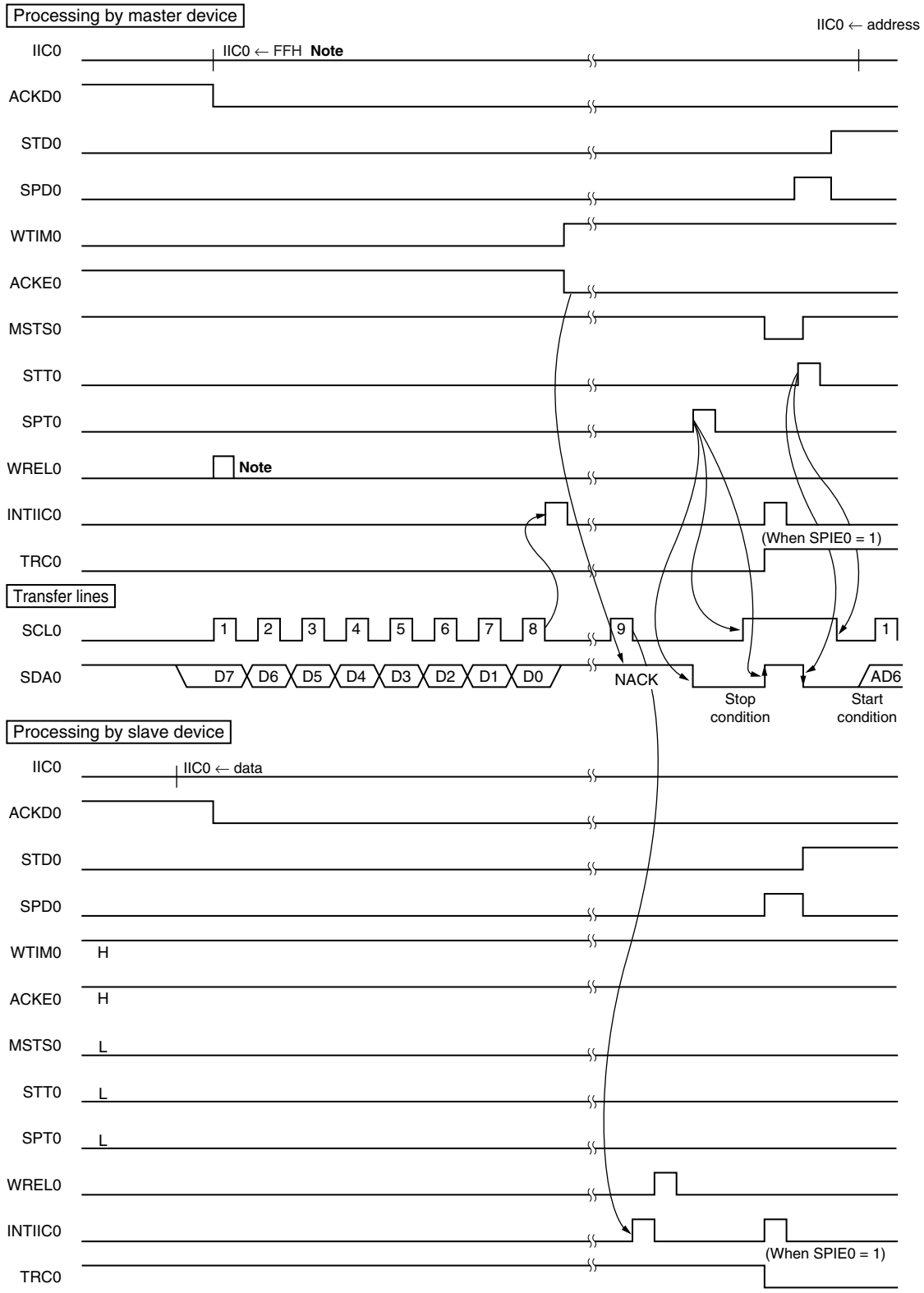
(2) Data



Note To cancel master wait, write “FFH” to IIC0 or set WREL0.

Figure 12-29. Example of Slave to Master Communication
(When 8-Clock and 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/3)

(3) Stop condition



Note To cancel master wait, write “FFH” to IIC0 or set WREL0.

CHAPTER 13 MULTIPLIER

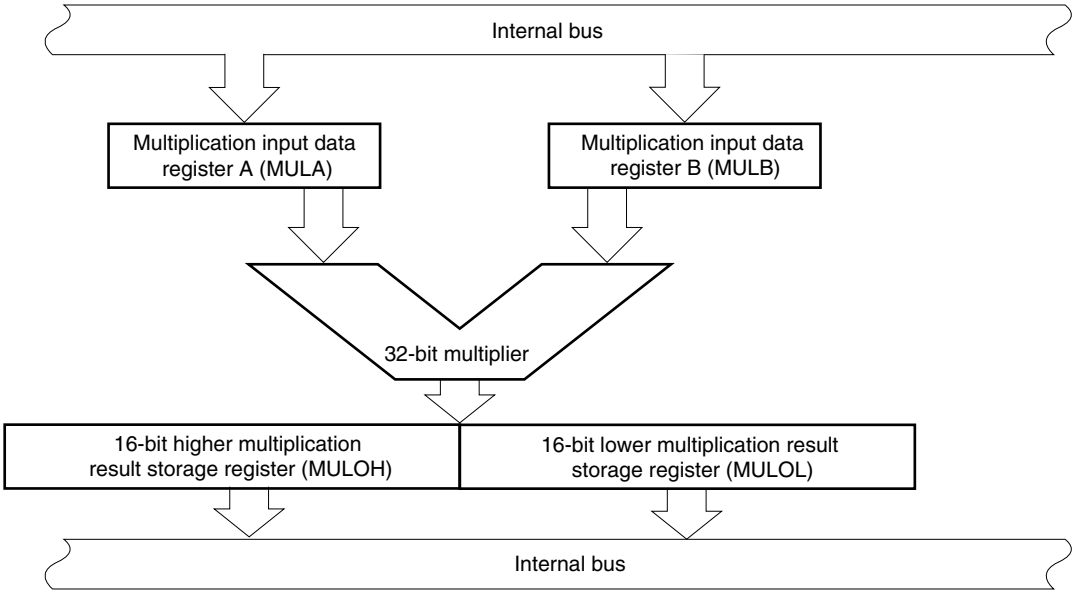
13.1 Functions of Multiplier

The multiplier has the following functions.

- Can execute calculation of 16 bits × 16 bits = 32 bits.

Figure 13-1 shows the block diagram of the multiplier.

Figure 13-1. Block Diagram of Multiplier



13.2 Configuration of Multiplier

(1) 16-bit higher multiplication result storage register and 16-bit lower multiplication result storage register (MULOH, MULOL)

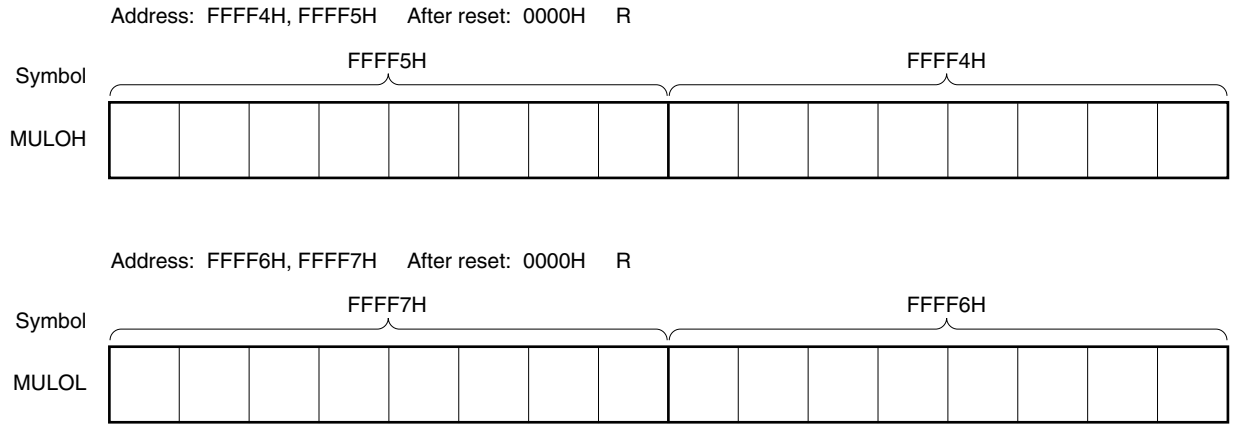
These two registers, MULOH and MULOL, are used to store a 32-bit multiplication result. The higher 16 bits of the multiplication result are stored in MULOH and the lower 16 bits, in MULOL, so that a total of 32 bits of the multiplication result can be stored.

These registers hold the result of multiplication after the lapse of one CPU clock.

MULOH and MULOL can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 13-2. Format of 16-bit higher multiplication result storage register and 16-bit lower multiplication result storage register (MULOH, MULOL)



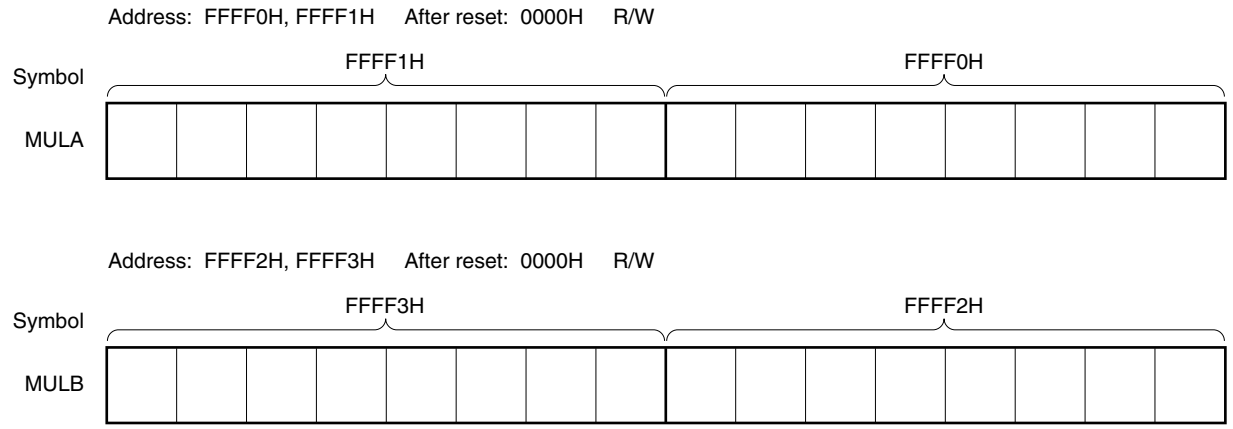
(2) Multiplication input data registers A, B (MULA, MULB)

These are 16-bit registers that store data for multiplication. The multiplier multiplies the values of MULA and MULB.

MULA and MULB can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 13-3. Format of Multiplication input data registers A, B (MULA, MULB)



13.3 Operation of Multiplier

The result of the multiplication can be obtained by storing the values in the MULA and MULB registers and then reading the MULOH and MULOL registers after waiting for 1 clock. The result can also be obtained after 1 clock or more has elapsed, even when fixing either of MULA or MULB and rewrite the other of these. The result can be read without problem, regardless of whether MULOH or MULOL is read in first.

A source example is shown below.

Example

```
MOVW        MULA, #1234H
MOVW        MULB, #5678H
NOP                             ; 1 clock wait.  Doesn't have to be NOP
MOVW        AX, MULOH           ; The result obtained on upper side
PUSH        AX
MOVW        AX, MULOL           ; The result obtained on lower side
```

CHAPTER 14 DMA CONTROLLER

The 78K0R/KE3 has an internal DMA (Direct Memory Access) controller.

Data can be automatically transferred between the peripheral hardware supporting DMA, SFRs, and internal RAM without via CPU.

As a result, the normal internal operation of the CPU and data transfer can be executed in parallel with transfer between the SFR and internal RAM, and therefore, a large capacity of data can be processed. In addition, real-time control using communication, timer, and A/D can also be realized.

14.1 Functions of DMA Controller

- Number of DMA channels: 2
- Transfer unit: 8 or 16 bits
- Maximum transfer unit: 1024 times
- Transfer type: 2-cycle transfer (One transfer is processed in 2 clocks and the CPU stops during that processing.)
- Transfer mode: Single-transfer mode
- Transfer request: Selectable from the following peripheral hardware interrupts
  - A/D converter
  - Serial interface (CIS00, CSI10, UART0, UART1, UART3, 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

14.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

Table 14-1. 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 register 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[Note].

This register is not automatically incremented but fixed to a specific value.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

DSAn can be read or written in 8-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 00H.

Note Except for address FFFFEH because the PMC register is allocated there.

Figure 14-1. 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 (FEF00H to FFEDFH in the case of the μPD78F1142) 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.

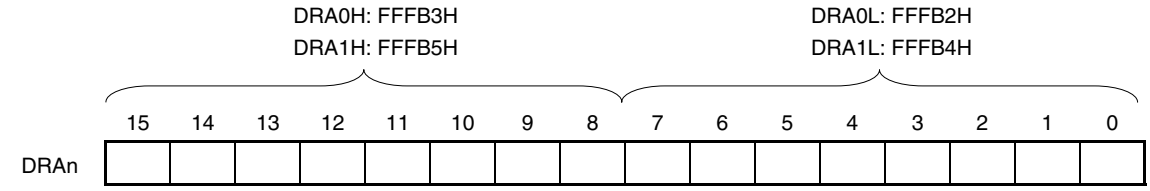
In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

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 14-2. 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 14-3. Format of DMA Byte Count Register n (DBCn)

Address: FFFB6H, FFFB7H (DBC0), FFFB8H, FFFB9H (DBC1) After reset: 0000H R/W

DBC0H: FFFB7H
DBC1H: FFFB9H

DBC0L: FFFB6H
DBC1L: FFFB8H

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

(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)

14.3 Registers to Controlling DMA Controller

DMA controller is controlled by the following registers.

- DMA mode control register n (DMCn)
- DMA operation control register n (DRCn)

Remark n: DMA channel number (n = 0, 1)

(1) DMA mode control register n (DMCn)

DMCn is a register that is used to set a transfer mode of DMA channel n. It is used to select a transfer direction, data size, setting of pending, and start source. Bit 7 (STGn) is a software trigger that starts DMA.
Rewriting bits 6, 5, and 3 to 0 of DMCn is prohibited during operation (when DSTn = 1).
DMCn can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 14-4. 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]	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	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.	

Note The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 values.

Remark n: DMA channel number (n = 0, 1)

Figure 14-4. Format of DMA Mode Control Register n (DMCn) (2/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

IFCn 3	IFCn 2	IFCn 1	IFCn 0	Selection of DMA stat source[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	UART0 reception 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	0	1	0	INTST3	UART3 transmission end interrupt
1	0	1	1	INTSR3	UART3 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 14-5. 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.
DMAC 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).
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1). When a software trigger (STGn) or the start source trigger set by IFCn3 to IFCn0 is input, DMA transfer is started. When DMA transfer is completed after that, this bit is automatically cleared to 0. Write 0 to this bit to forcibly terminate DMA transfer under execution.	

Caution The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENn flag is enabled only when DSTn = 0. When a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, therefore, set DSTn to 0 and then DENn to 0 (for details, refer to 14.5.5 Forced termination by software).

Remark n: DMA channel number (n = 0, 1)

14.4 Operation of DMA Controller

14.4.1 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, CBCn, and DMCn registers.

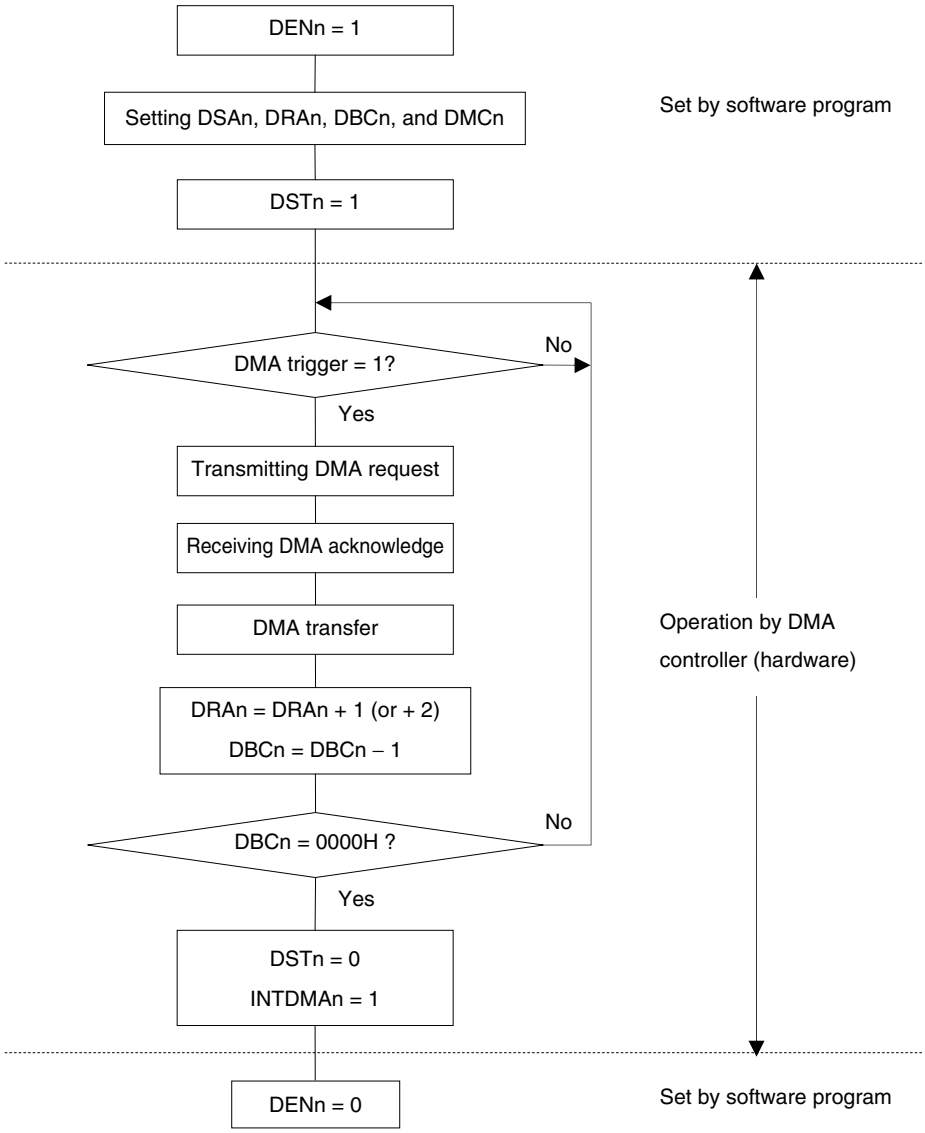
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.

<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 14-6. Operation Procedure



Remark n: DMA channel number (n = 0, 1)

14.4.2 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.

14.4.3 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)

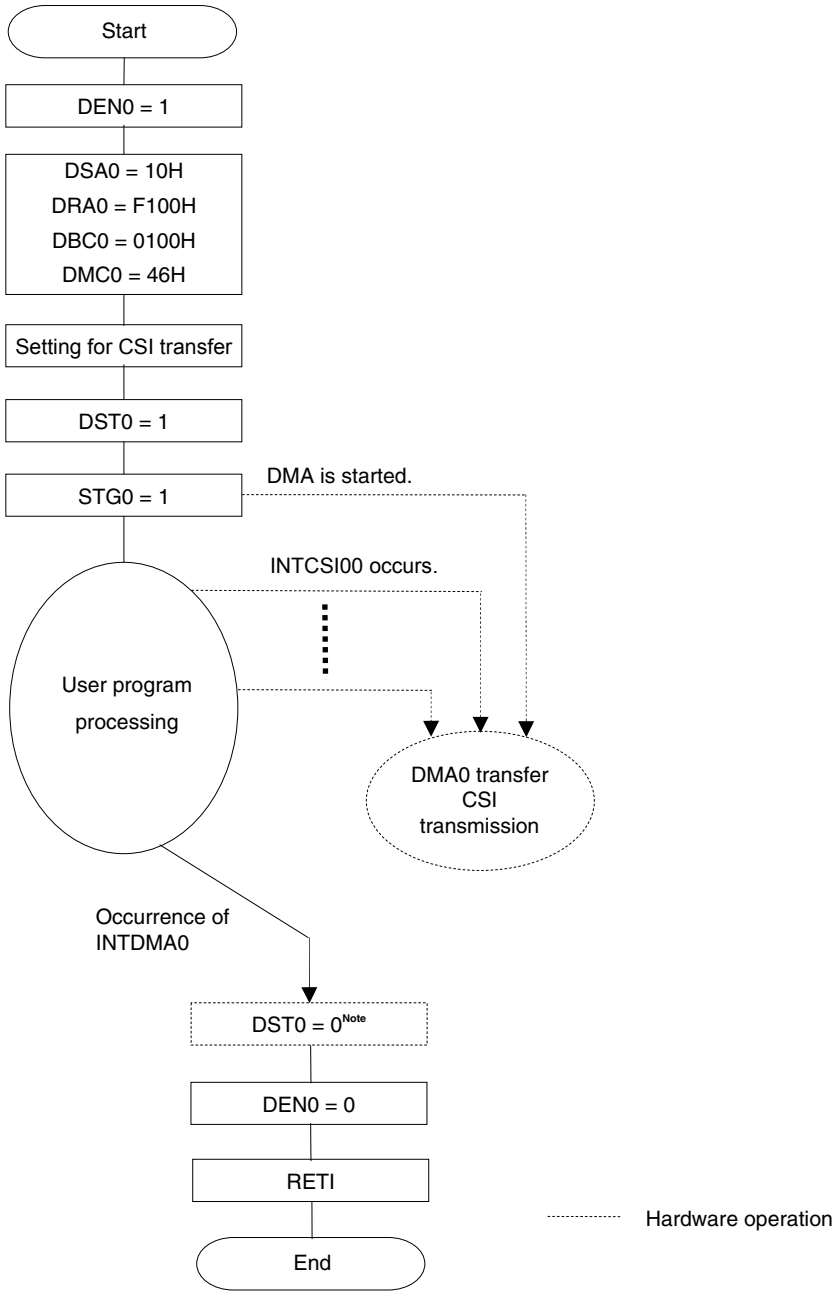
14.5 Example of Setting of DMA Controller

14.5.1 CSI consecutive transmission

A flowchart showing an example of setting for CSI consecutive transmission is shown below.

- Consecutive transmission of CSI00
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI00 (software trigger (STG0) only for the first start source)
- Interrupt of CSI00 is specified by IFC03 to IFC00 (bits 3 to 0 of the DMC0 register) = 0110B.
- Transfers FF100H to FF1FFH (256 bytes) of RAM to FFF10H of the transmit buffer (SIO00) of CSI.

Figure 14-7. 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 **14.5.5 Forced termination by software**).

The fist trigger for consecutive transmission is not started by the interrupt of CSI. Start it by a software trigger.

CSI transmission of the second time and onward is automatically executed.

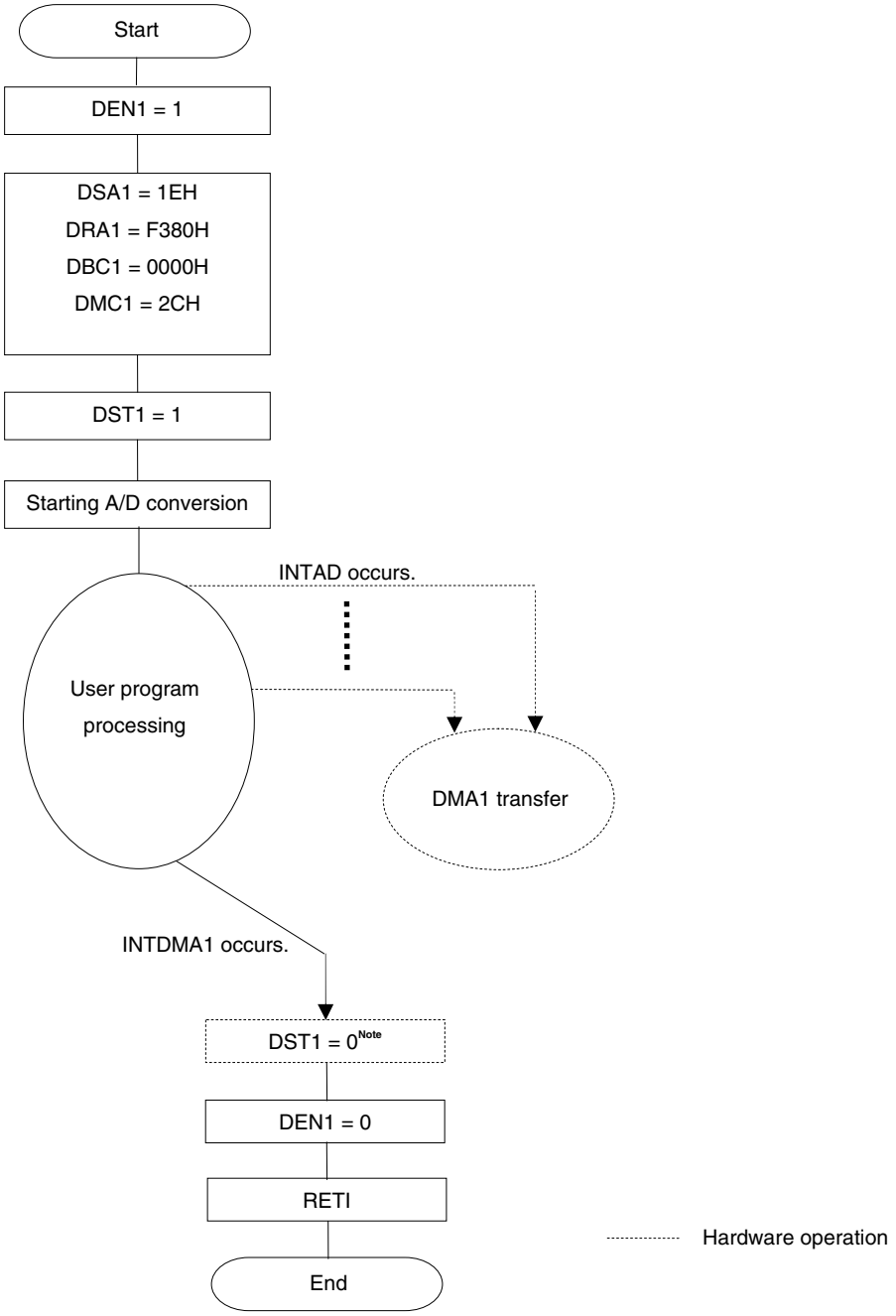
The DMA interrupt (INTDMA0) is generated as soon as the last data has been written to the transmit buffer. At this point, the last data of CSI is being transmitted. To start DMA transfer again, therefore, wait until transfer of CSI is completed.

14.5.2 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 2048 bytes of FF380H to FFB7FH of RAM.

Figure 14-8. Example of Setting of Consecutively Capturing A/D Conversion Results



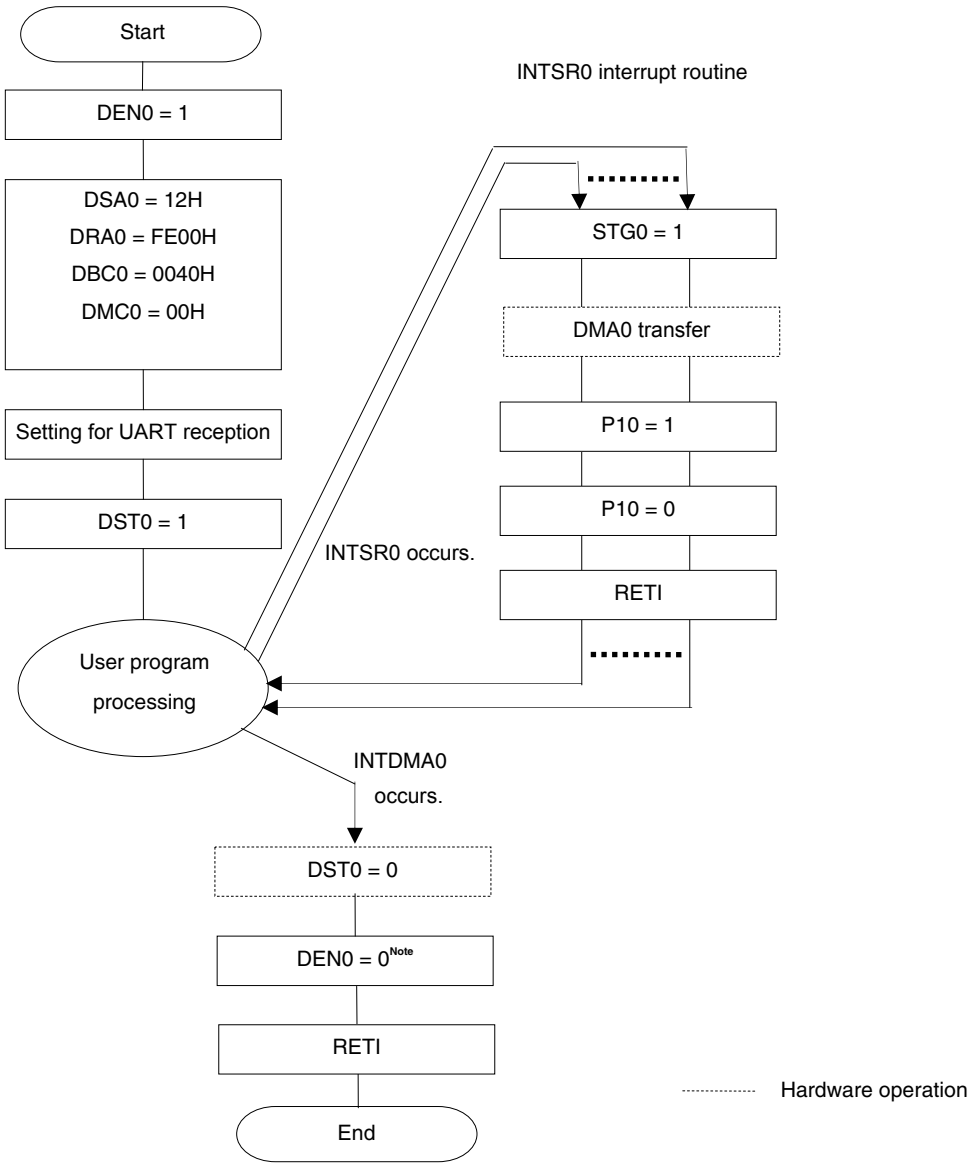
Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set DST1 to 0 and then DEN1 to 0 (for details, refer to **14.5.5 Forced termination by software**).

14.5.3 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 14-9. 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 **14.5.5 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.

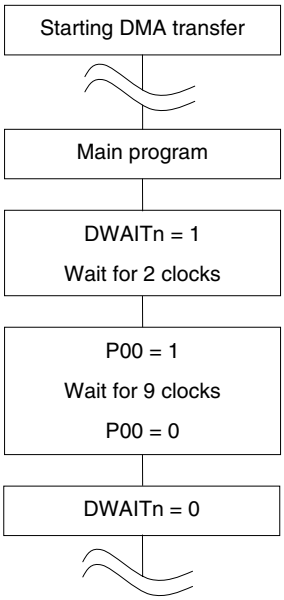
14.5.4 Holding DMA transfer pending by DWAITn

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 DWAITn to 1.

To output a pulse with a width of 10 clocks of the operating frequency from the P00 pin, for example, the clock width increases to 12 if a DMA transfer is started midway. In this case, the DMA transfer can be held pending by setting DWAITn to 1.

After setting DWAITn to 1, it takes two clocks until a DMA transfer is held pending.

Figure 14-10. Example of Setting for Holding DMA Transfer Pending by DWAITn



Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

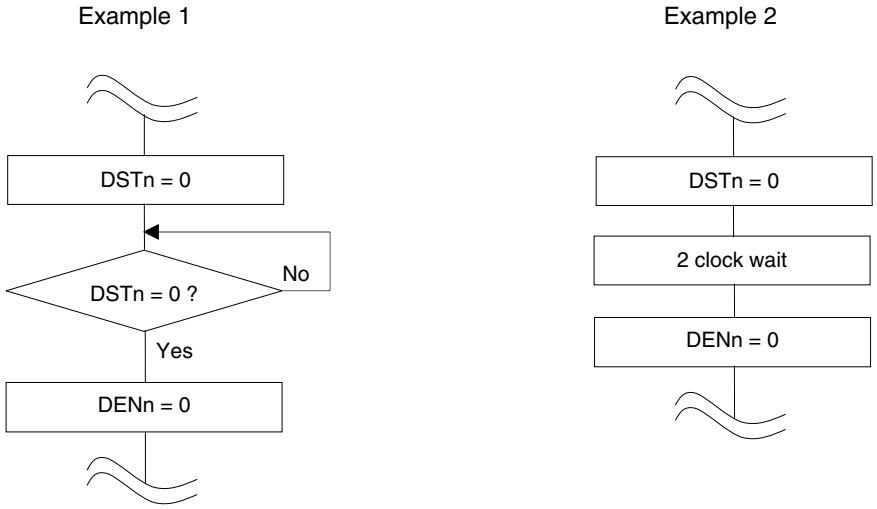
14.5.5 Forced termination by software

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.

- 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.

Figure 14-11. Forced Termination of DMA Transfer



Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

14.6 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. When the requests from either of the DMA channels are successively generated in a short period [Note], they are successively transferred, and on completion of that, the requests from the other DMA channel are executed. In this case, one or tow instructions are executed between the first DMA transfer and next DMA transfer.

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.

Note The short period refers to a period of eight or fewer CPU clocks. The relationship between the lengths of clock period and DMA operations is as follows.

1 clock period:	Setting disabled DMA request cannot be accepted.
2 to 4 clock period:	DMA transfer of the channel where requests are successively generated is executed.
5 to 8 clock period:	Whether DMA transfer of the channel where requests are successively generated is executed or DMA requests from the other channel are executed depends on the number of times CPU instructions are executed.

(2) DMA response time

The response time of DMA transfer is as follows.

Table 14-2. Response Time of DMA Transfer

	Minimum Time	Maximum Time
Response time	4 clocks	10 clocks

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

In the following cases, however, DMA transfer may be delayed further. The number of clocks by which DMA transfer is delayed differs depending on the condition.

- Instruction execution by RAM
- Instruction execution by external memory
- If wait cycle is inserted when external memory is accessed
- Execution of DMA pending instruction

(3) Operation in standby mode

The DMA controller operates as follows in the standby mode.

Table 14-3. 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 &!addr16
- 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, and 8-bit manipulation instructions with operands including ES registers

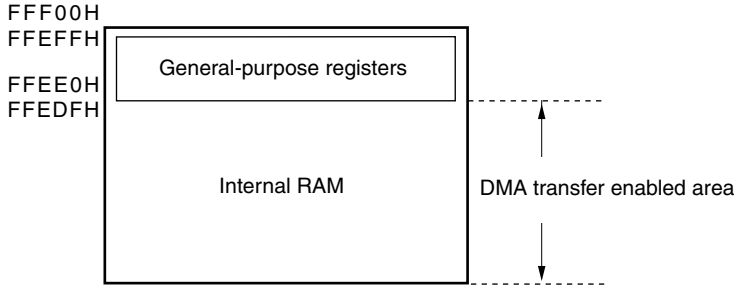
(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified

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 15 INTERRUPT FUNCTIONS

15.1 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 15-1**.

A standby release signal is generated and STOP and HALT modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

External: 13, internal: 25

(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.

15.2 Interrupt Sources and Configuration

The 78K0R/KE3 has a total of 39 interrupt sources including maskable interrupts and software interrupts. In addition, they also have up to five reset sources (see **Table 15-1**).

Table 15-1. Interrupt Source List (1/2)

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	INTST3	End of UART3 transmission	Internal	0014H	(A)
	9	INTSR3	End of UART3 reception		0016H	
	10	INTSRE3	UART3 reception error occurrence		0018H	
	11	INTDMA0	End of DMA0 transfer		001AH	
	12	INTDMA1	End of DMA1 transfer		001CH	
	13	INTST0 /INTCSI00	End of UART0 transmission/ end of CSI00 communication		001EH	
	14	INTSR0	End of UART0 reception		0020H	
<R>	15	INTSRE0	UART0 reception error occurrence		0022H	
	16	INTST1 /INTCSI10 /INTIIC10	End of UART1 transmission/ end of CSI10 communication/ end of IIC10 communication		0024H	
	17	INTSR1	End of UART1 reception		0026H	
<R>	18	INTSRE1	UART1 reception error occurrence		0028H	
	19	INTIIC0	End of IIC0 communication		002AH	
	20	INTTM00	End of timer channel 0 count or capture		002CH	
	21	INTTM01	End of timer channel 1 count or capture		002EH	
	22	INTTM02	End of timer channel 2 count or capture		0030H	
	23	INTTM03	End of timer 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 37 indicates the lowest priority.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 15-1.

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 15-1. Interrupt Source List (2/2)

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	(B)
	28	INTTM04	End of timer channel 4 count or capture	Internal	0042H	(A)
	29	INTTM05	End of timer channel 5 count or capture		0044H	
	30	INTTM06	End of timer channel 6 count or capture		0046H	
	31	INTTM07	End of timer channel 7 count or capture		0048H	
	32	INTP6	Pin input edge detection	External	004AH	(B)
	33	INTP7			004CH	
	34	INTP8			004EH	
	35	INTP9			0050H	
	36	INTP10			0052H	
	37	INTP11			0054H	
Software	–	BRK	Execution of BRK instruction	–	007EH	(C)
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 37 indicates the lowest priority.

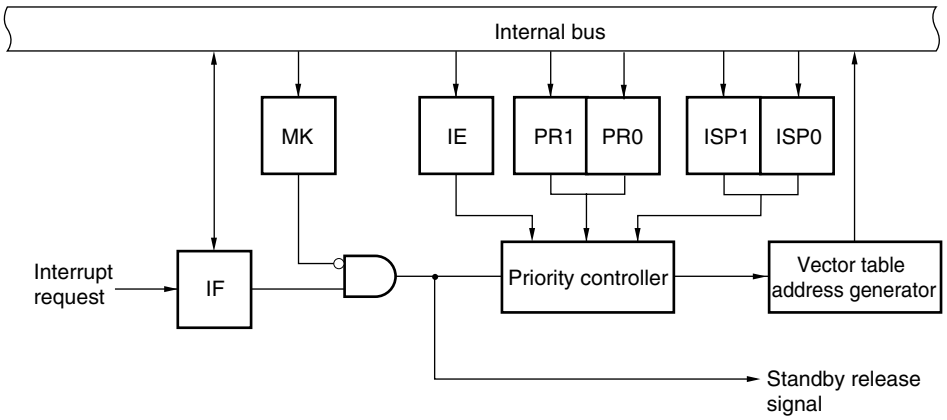
2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 15-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

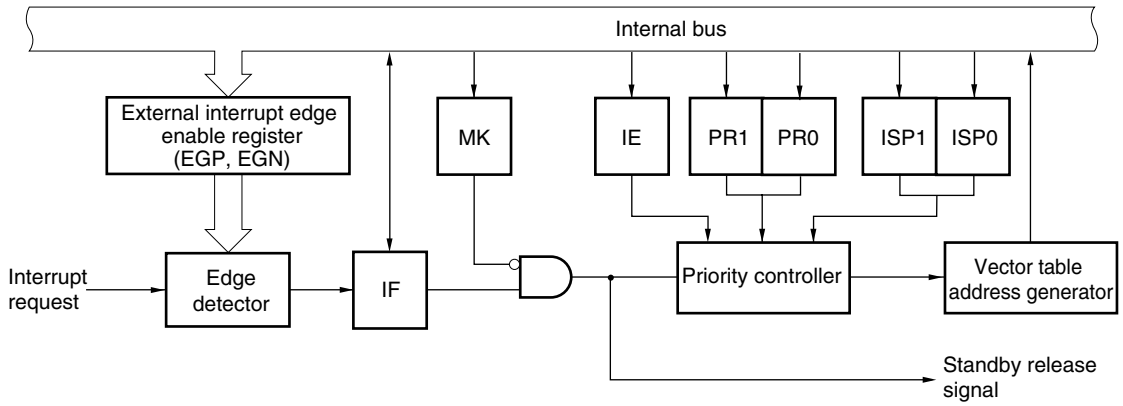
4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 15-1. Basic Configuration of Interrupt Function

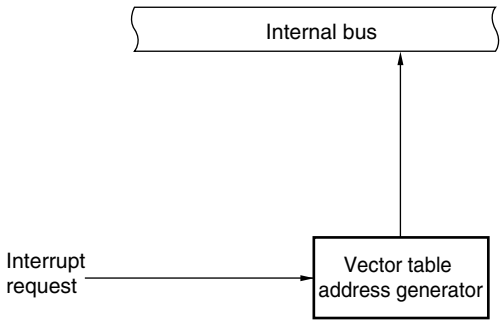
(A) Internal maskable interrupt



(B) External maskable interrupt



(C) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

15.3 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 15-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 15-2. 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	
INTST3	STIF3	IF0H	STMK3	MK0H	STPR03, STPR13	PR00H, PR10H
INTSR3	SRIF3		SRMK3		SRPR03, SRPR13	
INTSRE3	SREIF3		SREMK3		SREPR03, SREPR13	
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10	
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11	
INTST0 Note	STIF0 Note		STMK0 Note		STPR00, STPR10 Note	
INTCSI00 Note	CSIIF00 Note		CSIMK00 Note		CSIPR000, CSIPR100 Note	
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10	
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10	

Note 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.

Table 15-2. Flags Corresponding to Interrupt Request Sources (2/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTST1 Note	STIF1 Note	IF1L	STMK1 Note	MK1L	STPR01, STPR11 Note	PR01L, PR11L
INTCSI10 Note	CSIIF10 Note		CSIMK10 Note		CSIPR010, CSIPR110 Note	
INTIIC10 Note	IICIF10 Note		IICMK10 Note		IICPR010, IICPR110 Note	
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11	
INTSRE1	SREIF1		SREMK1		SREPR01, SREPR11	
INTIIC0	IICIF0		IICMK0		IICPR00, IICPR10	
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101	
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102	
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103	
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1	
INTRTCI	RTCIIF		RTCIMK		RTCIPR0, RTCIPR1	
INTKR	KRIF		KRMK		KRPR0, KRPR1	
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104	
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106	
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107	
INTP6	PIF6		PMK6		PPR06, PPR16	
INTP7	PIF7		PMK7		PPR07, PPR17	
INTP8	PIF8		PMK8		PPR08, PPR18	
INTP9	PIF9		PMK9		PPR09, PPR19	
INTP10	PIF10		PMK10		PPR010, PPR110	
INTP11	PIF11	IF2H	PMK11	MK2H	PPR011, PPR111	PR02H, PR12H

Note 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.

(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 15-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

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	CSIIF00 STIF0	DMAIF1	DMAIF0	SREIF3	SRIF3	STIF3

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	IICIF0	SREIF1	SRIF1	CSIIF10 IICIF10 STIF1

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
IF1H	TMIF04	0	0	0	KRIF	RTCIIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	PIF10	PIF9	PIF8	PIF7	PIF6	TMIF07	TMIF06	TMIF05

Address: FFFD1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
IF2H	0	0	0	0	0	0	0	PIF11

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. Be sure to clear bits 4 to 6 of IF1H and bits 1 to 7 of IF2H to 0.

2. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

3. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H can be set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, MK1L and MK1H, and MK2L and MK2H are combined to form 16-bit registers MK0, MK1, and MK2, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 15-3. 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	CSIMK00 STMK0	DMAMK1	DMAMK0	SREMK3	SRMK3	STMK3

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	IICMK0	SREMK1	SRMK1	CSIMK10 IICMK10 STMK1

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
MK1H	TMMK04	1	1	1	KRMK	RTCIMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	PMK10	PMK9	PMK8	PMK7	PMK6	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
MK2H	1	1	1	1	1	1	1	PMK11

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bits 4 to 6 of MK1H and bits 1 to 7 of MK2H to 1.

(3) Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority level.
A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L, or 2H).
PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H can be set by a 1-bit or 8-bit memory manipulation instruction. If PR00L and PR00H, PR01L and PR01H, PR02L and PR02H, PR10L and PR10H, PR11L and PR11H, and PR12L and PR12H are combined to form 16-bit registers PR00, PR01, PR02, PR10, PR11, and PR12, they can be set by a 16-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 15-4. 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	CSIPR000 STPR00	DMAPR01	DMAPR00	SREPR03	SRPR03	STPR03

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10	SRPR10	CSIPR100 STPR10	DMAPR11	DMAPR10	SREPR13	SRPR13	STPR13

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICPR00	SREPR01	SRPR01	CSIPR010 IICPR010 STPR01

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICPR10	SREPR11	SRPR11	CSIPR110 IICPR110 STPR11

Figure 15-4. 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	1	1	1	KRPR0	RTCIPR0	RTCPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
PR11H	TMPR104	1	1	1	KRPR1	RTCIPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	PPR010	PPR09	PPR08	PPR07	PPR06	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	PPR110	PPR19	PPR18	PPR17	PPR16	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
PR02H	1	1	1	1	1	1	1	PPR011

Address: FFFDDH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
PR12H	1	1	1	1	1	1	1	PPR111

XXPR1X	XXPR0X	Priority level selection
0	0	Specify level 0 (high priority level)
0	1	Specify level 1
1	0	Specify level 2
1	1	Specify level 3 (low priority level)

Caution Be sure to set bits 4 to 6 of PR01H and PR11H to 1.
Be sure to set bits 1 to 7 of PR02H and PR12H to 1.

(4) External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)

These registers specify the valid edge for INTP0 to INTP11.

EGP0, EGP1, EGN0, and EGN1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 15-5. Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP11	EGP10	EGP9	EGP8

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN11	EGN10	EGN9	EGN8

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 11)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 15-3 shows the ports corresponding to EGPn and EGNn.

Table 15-3. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P50	INTP1
EGP2	EGN2	P51	INTP2
EGP3	EGN3	P30	INTP3
EGP4	EGN4	P31	INTP4
EGP5	EGN5	P16	INTP5
EGP6	EGN6	P140	INTP6
EGP7	EGN7	P141	INTP7
EGP8	EGN8	P74	INTP8
EGP9	EGN9	P75	INTP9
EGP10	EGN10	P76	INTP10
EGP11	EGN11	P77	INTP11

Caution Select the port mode by clearing EGPn and EGNn to 0 because an edge may be detected when the external interrupt function is switched to the port function.

Remark n = 0 to 11

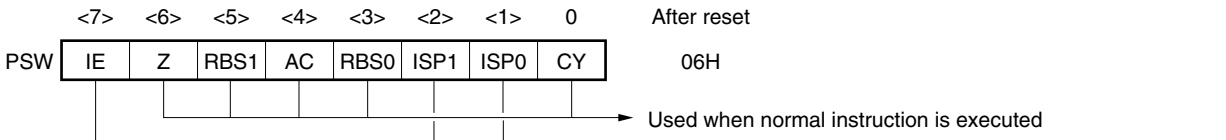
(5) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are transferred to the ISP0 and ISP1 flags. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 15-6. 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

15.4 Interrupt Servicing Operations

15.4.1 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 15-4 below.

For the interrupt request acknowledgment timing, see **Figures 15-8** and **15-9**.

Table 15-4. 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.

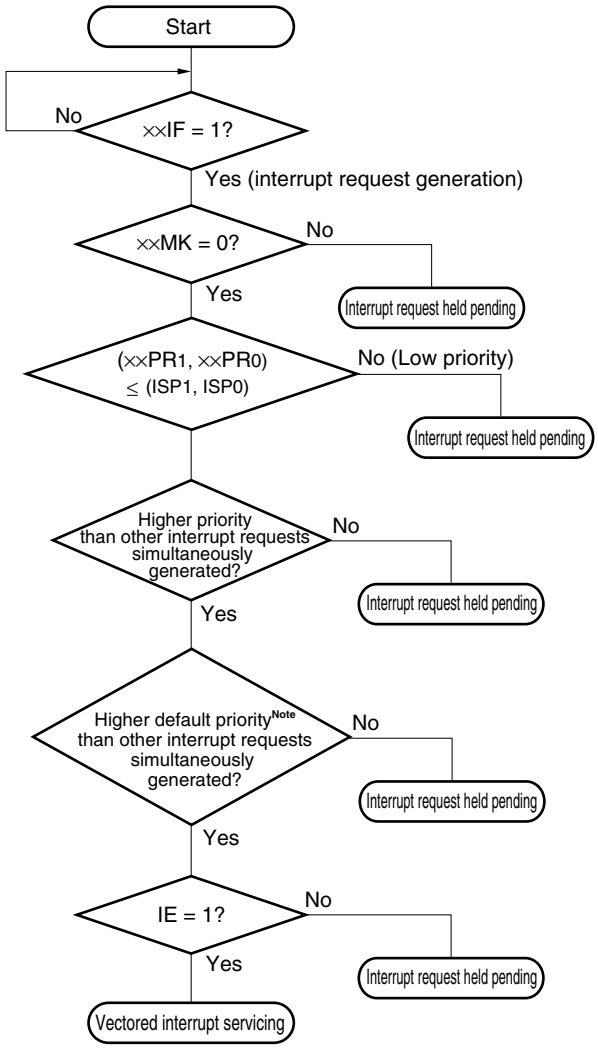
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 15-7 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.

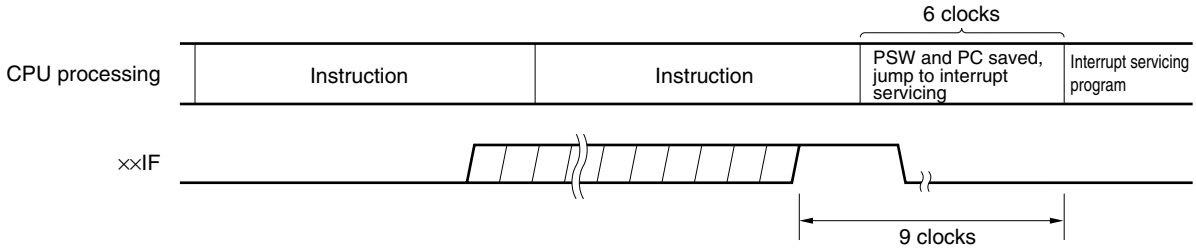
Figure 15-7. Interrupt Request Acknowledgment Processing Algorithm



××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 15-6**)

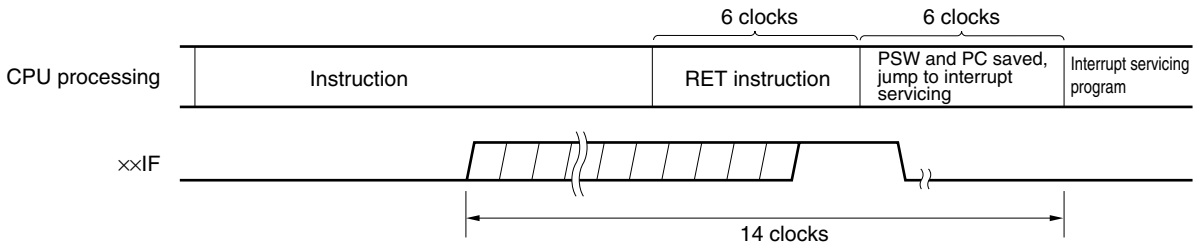
Note For the default priority, refer to **Table 15-1 Interrupt Source List**.

Figure 15-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

Figure 15-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

15.4.2 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.

15.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 15-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 15-10 shows multiple interrupt servicing examples.

Table 15-5. 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)

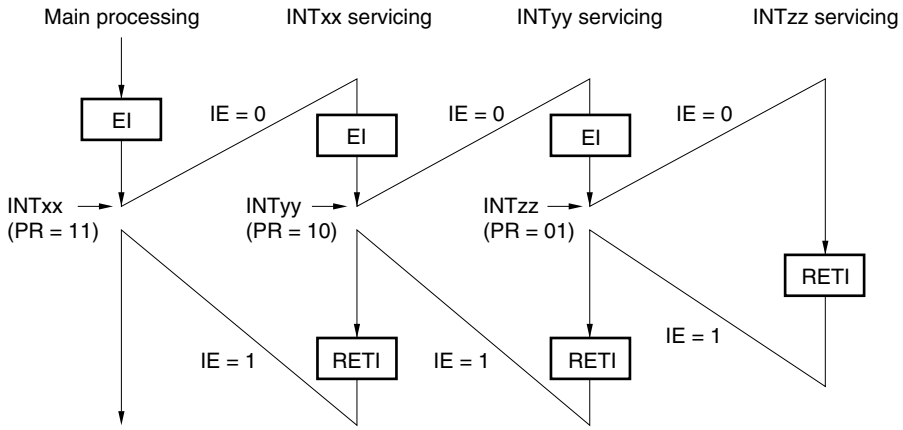
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 1 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

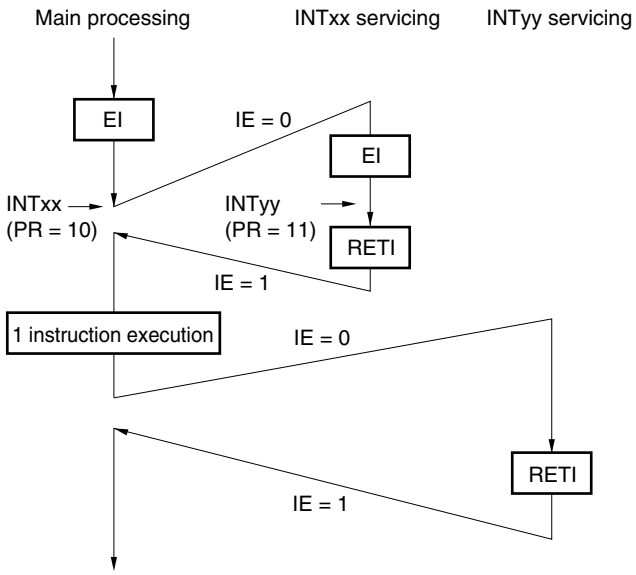
Figure 15-10. Examples of Multiple Interrupt Servicing (1/2)

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

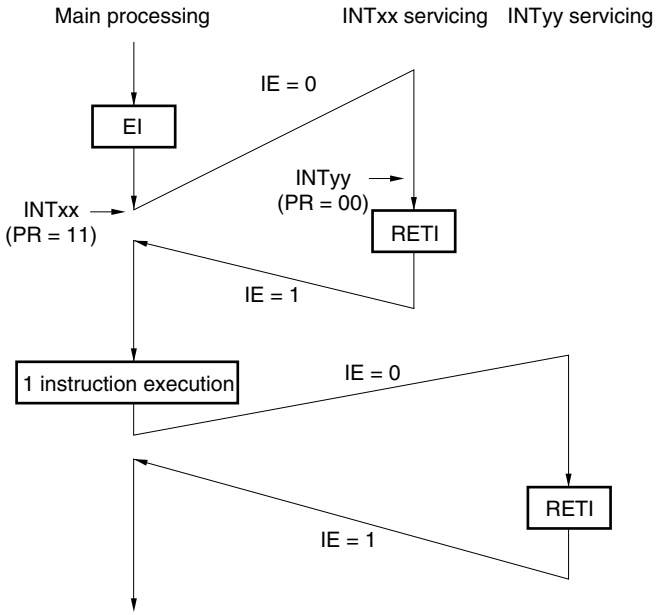


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

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 1 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 15-10. 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 1 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

15.4.4 Interrupt request hold

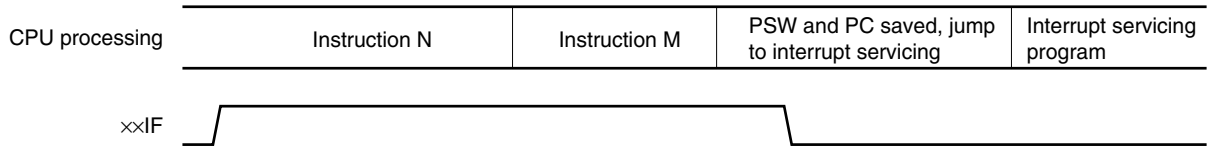
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr8
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- 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 15-11 shows the timing at which interrupt requests are held pending.

Figure 15-11. 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 16 KEY INTERRUPT FUNCTION

16.1 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 16-1. 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.

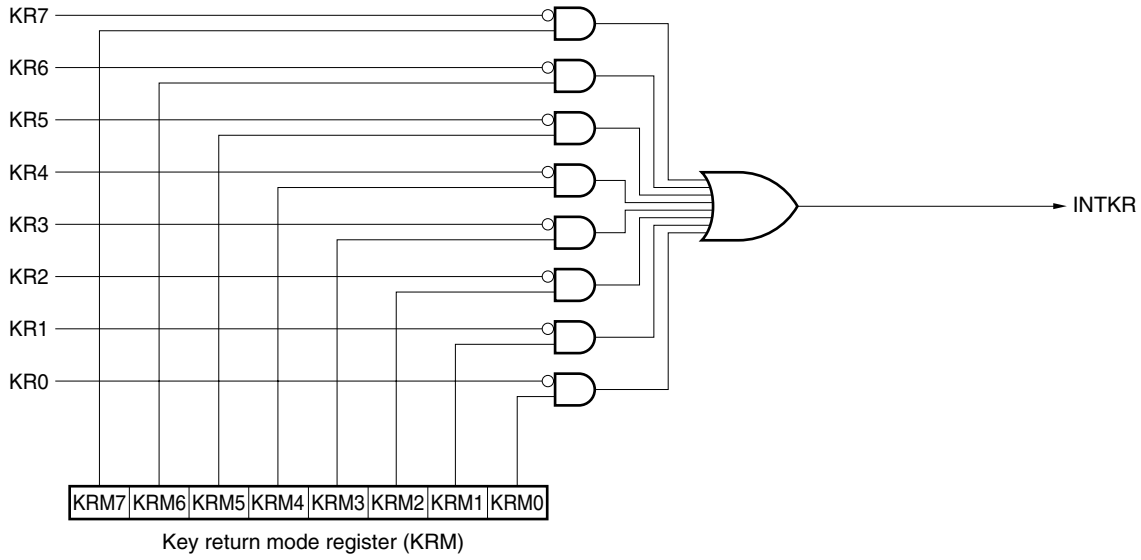
16.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 16-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return mode register (KRM)

Figure 16-1. Block Diagram of Key Interrupt



16.3 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 16-2. 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. If KRM is changed, the interrupt request flag may be set. Therefore, disable interrupts and then change the KRM register. Clear the interrupt request flag and enable interrupts.

3. The bits not used in the key interrupt mode can be used as normal ports.

Remark n = 0 to 7

CHAPTER 17 STANDBY FUNCTION

17.1 Standby Function and Configuration

17.1.1 Standby function

The standby function reduces the operating current of the system, and the following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator, internal high-speed oscillator, or subsystem clock oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and internal high-speed oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the CPU is operating on the main system clock. The STOP mode cannot be set while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.

3. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

4. It can be selected by the option byte whether the internal low-speed oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 22 OPTION BYTE.

17.1.2 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 17-1. 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		
									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.

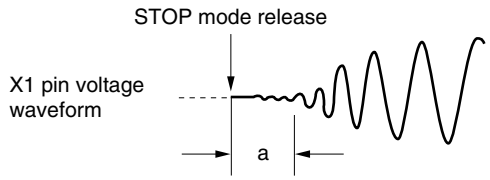
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. 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 f_X: X1 clock oscillation frequency

(2) Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time when the STOP mode is released.

When the X1 clock is selected as the CPU clock, the operation waits for the time set using OSTS after the STOP mode is released.

When the internal high-speed oscillation clock is selected as the CPU clock, confirm with OSTC that the desired oscillation stabilization time has elapsed after the STOP mode is released. The oscillation stabilization time can be checked up to the time set using OSTC.

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

Reset signal generation sets this register to 07H.

Figure 17-2. 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.

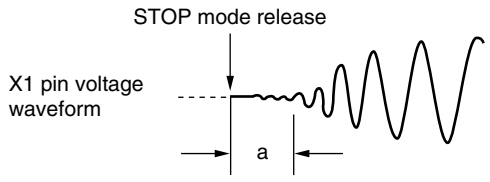
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

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

17.2 Standby Function Operation

17.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock, internal high-speed oscillation clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Table 17-1. 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})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Operation continues (cannot be stopped)	Status before HALT mode was set is retained	
		f_X	Status before HALT mode was set is retained	Operation continues (cannot be stopped)	Cannot operate
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained		
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Oscillates • WTON = 1 and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Flash memory			Operable in low-current consumption mode		
RAM			Operation stopped. However, status before HALT mode was set is retained at voltage higher than POC detection voltage.		
Port (latch)			Status before HALT mode was set is retained		
Timer array unit (TAU)			Operable		
Real-time counter (RTC)					
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Operates • WTON = 1 and WDSTBYON = 0: Stops		
Clock output/buzzer output			Operable		
A/D converter					
Serial array unit (SAU)					
Serial interface (IIC0)					
Multiplier			Operation stopped		
DMA controller			Operable		
Power-on-clear function					
Low-voltage detection function					
External interrupt					

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

Table 17-1. 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}	Status before HALT mode was set is retained
		f_X	
		f_{EX}	Operates or stops by external clock input
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Oscillates • WTON = 1 and WDSTBYON = 0: Stops
CPU			Operation stopped
Flash memory			Operable in low-current consumption mode
RAM			Operation stopped. However, status before HALT mode was set is retained at voltage higher than POC detection voltage.
Port (latch)			Status before HALT mode was set is retained
Timer array unit (TAU)			Operable
Real-time counter (RTC)			
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Operates • WTON = 1 and WDSTBYON = 0: Stops
Clock output/buzzer output			Operable
A/D converter			Cannot operate
Serial array unit (SAU)			Operable
Serial interface (IIC0)			Cannot operate
Multiplier			Operation stopped
DMA controller			Operable
Power-on-clear function			
Low-voltage detection function			
External interrupt			

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

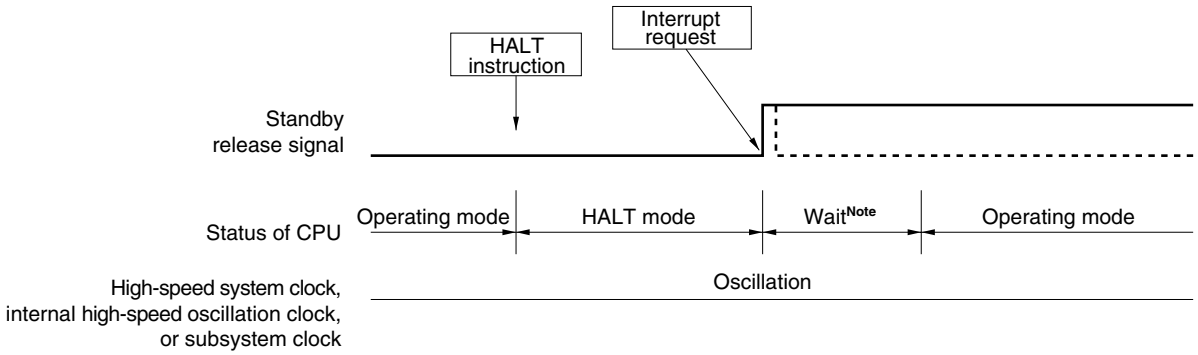
(2) HALT mode release

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 17-3. HALT Mode Release by Interrupt Request Generation



<R> **Note** The wait time is as follows:

- When vectored interrupt servicing is carried out: 10 to 12 clocks
- When vectored interrupt servicing is not carried out: 5 or 6 clocks

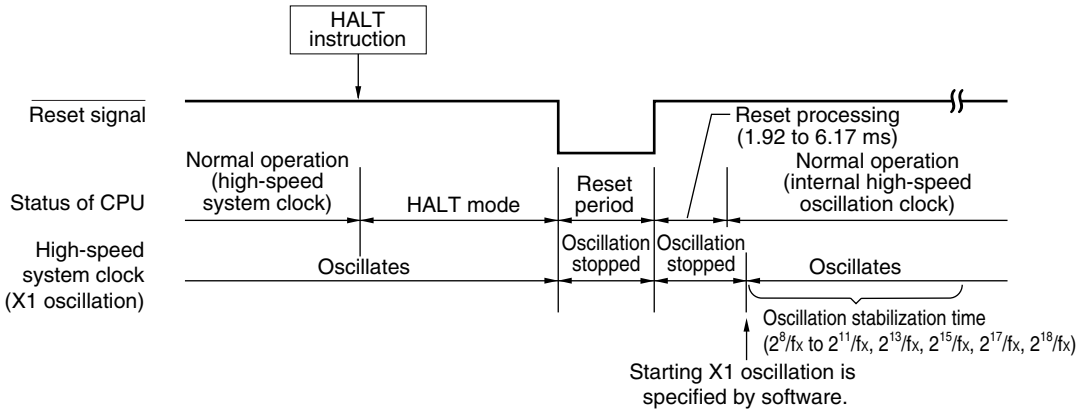
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

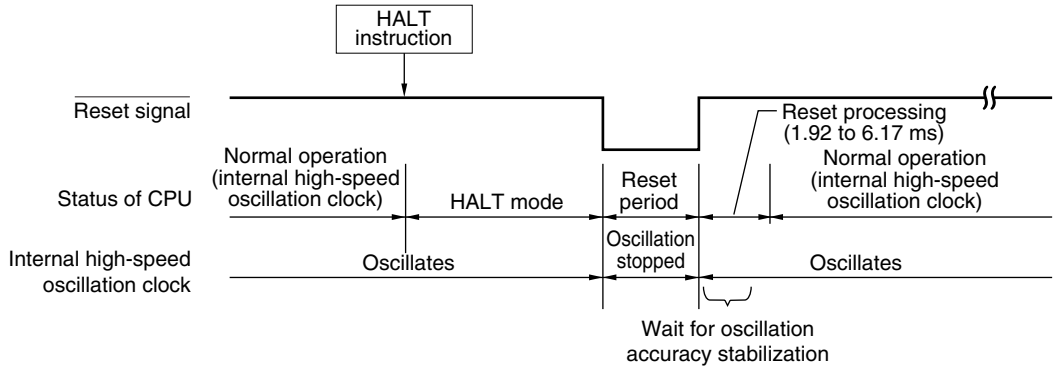
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 17-4. HALT Mode Release by Reset

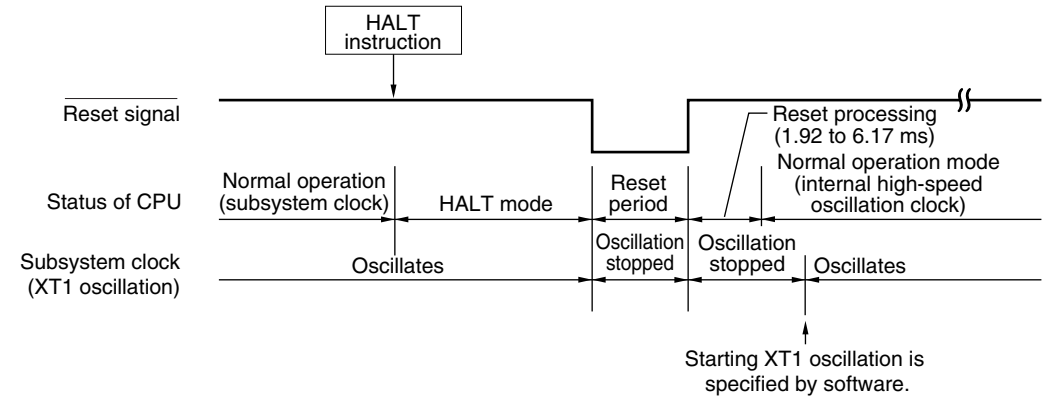
(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock is used as CPU clock



(3) When subsystem clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

17.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the main system clock.

Caution Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

The operating statuses in the STOP mode are shown below.

Table 17-2. 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 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Oscillates • WTON = 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 stopped		
Real-time counter (RTC)			Operable		
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WTON) of option byte (000C0H) • WTON = 0: Stops • WTON = 1 and WDSTBYON = 1: Operates • WTON = 1 and WDSTBYON = 0: Stops		
Clock output/buzzer output			Operable only when subsystem clock is selected as the count clock		
A/D converter			Operation stopped		
Serial array unit (SAU)					
Serial interface (IIC0)					
Multiplier					
DMA controller					
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					

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

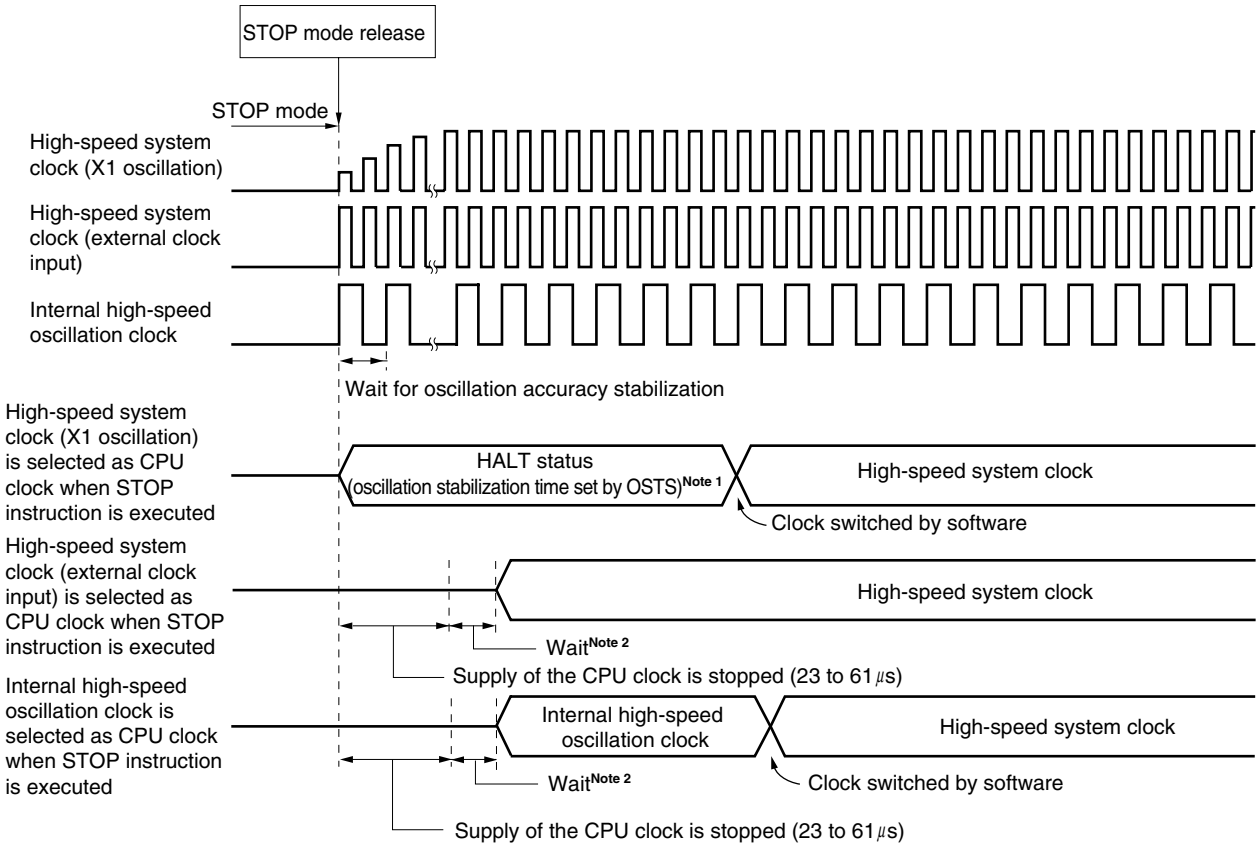
Cautions 1. To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.

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 next execution of the STOP instruction. Before changing the CPU clock from the internal high-speed oscillation clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).

(2) STOP mode release

Figure 17-5. Operation Timing When STOP Mode Is Released



Notes 1. When the oscillation stabilization time set by OSTS is equal to or shorter than 61 μs, the HALT status is retained to a maximum of "61μs + wait time."

<R> **2.** The wait time is as follows:

- When vectored interrupt servicing is carried out: 10 to 12 clocks
- When vectored interrupt servicing is not carried out: 5 or 6 clocks

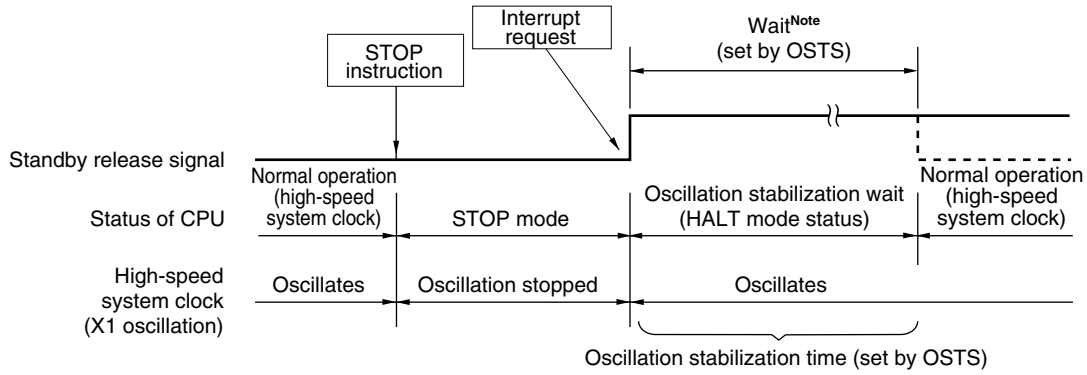
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

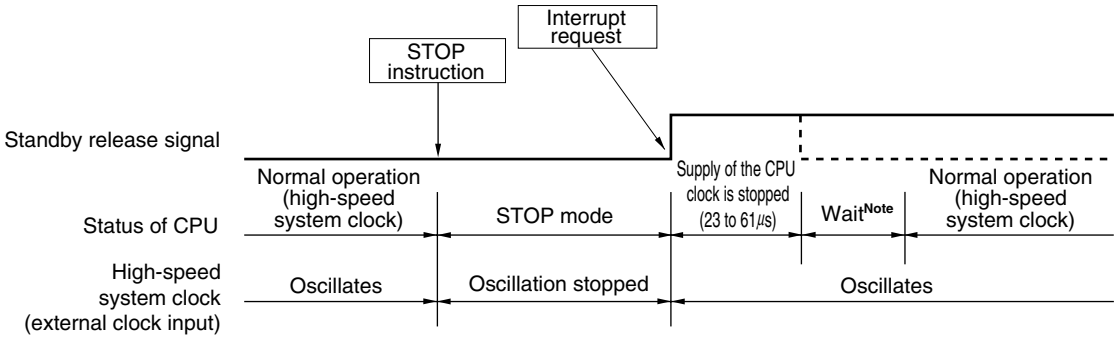
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 17-6. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed system clock (X1 oscillation) is used as CPU clock



(2) When high-speed system clock (external clock input) is used as CPU clock



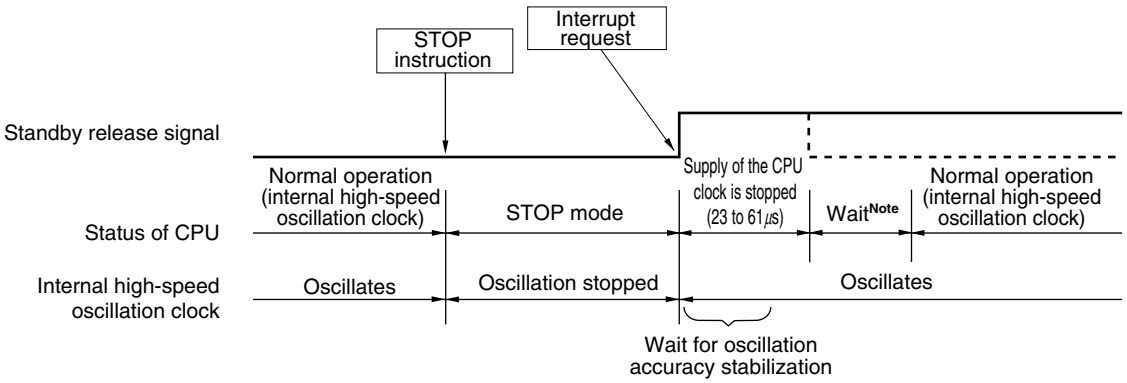
<R> **Note** The wait time is as follows:

- When vectored interrupt servicing is carried out: 10 to 12 clocks
- When vectored interrupt servicing is not carried out: 5 or 6 clocks

Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 17-6. STOP Mode Release by Interrupt Request Generation (2/2)

(3) When internal high-speed oscillation clock is used as CPU clock



<R> **Note** The wait time is as follows:

- When vectored interrupt servicing is carried out: 10 to 12 clocks
- When vectored interrupt servicing is not carried out: 5 or 6 clocks

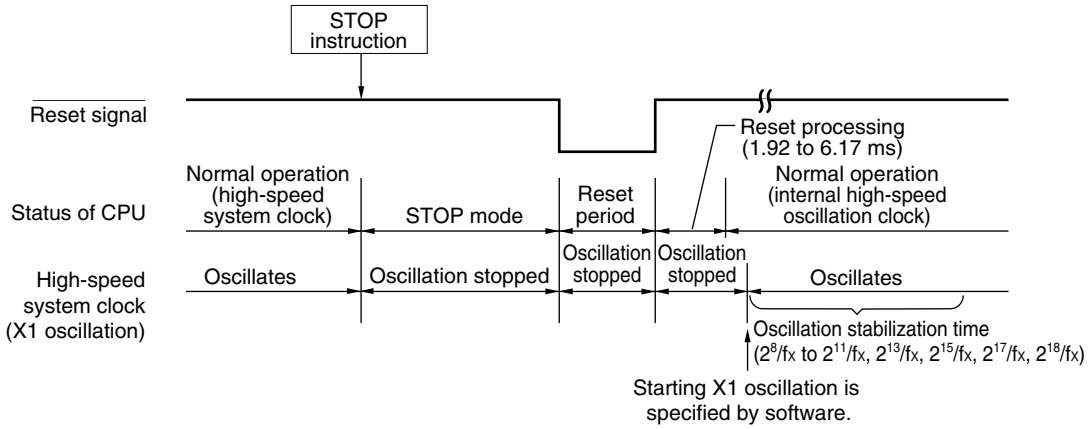
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

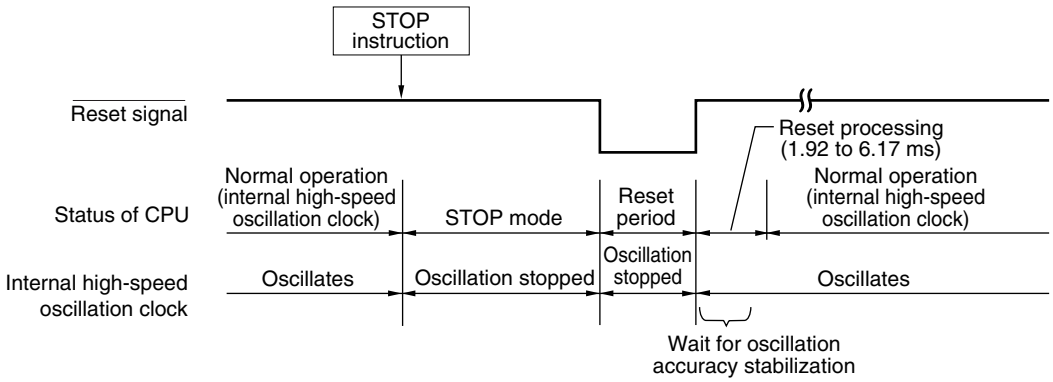
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 17-7. STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

CHAPTER 18 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
<R> (4) Internal reset by comparison of supply voltage of the low-voltage detector (LVI) or input voltage (EXLVI) from external input pin, and detection voltage
(5) Internal reset by execution of illegal instruction[Note]

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.

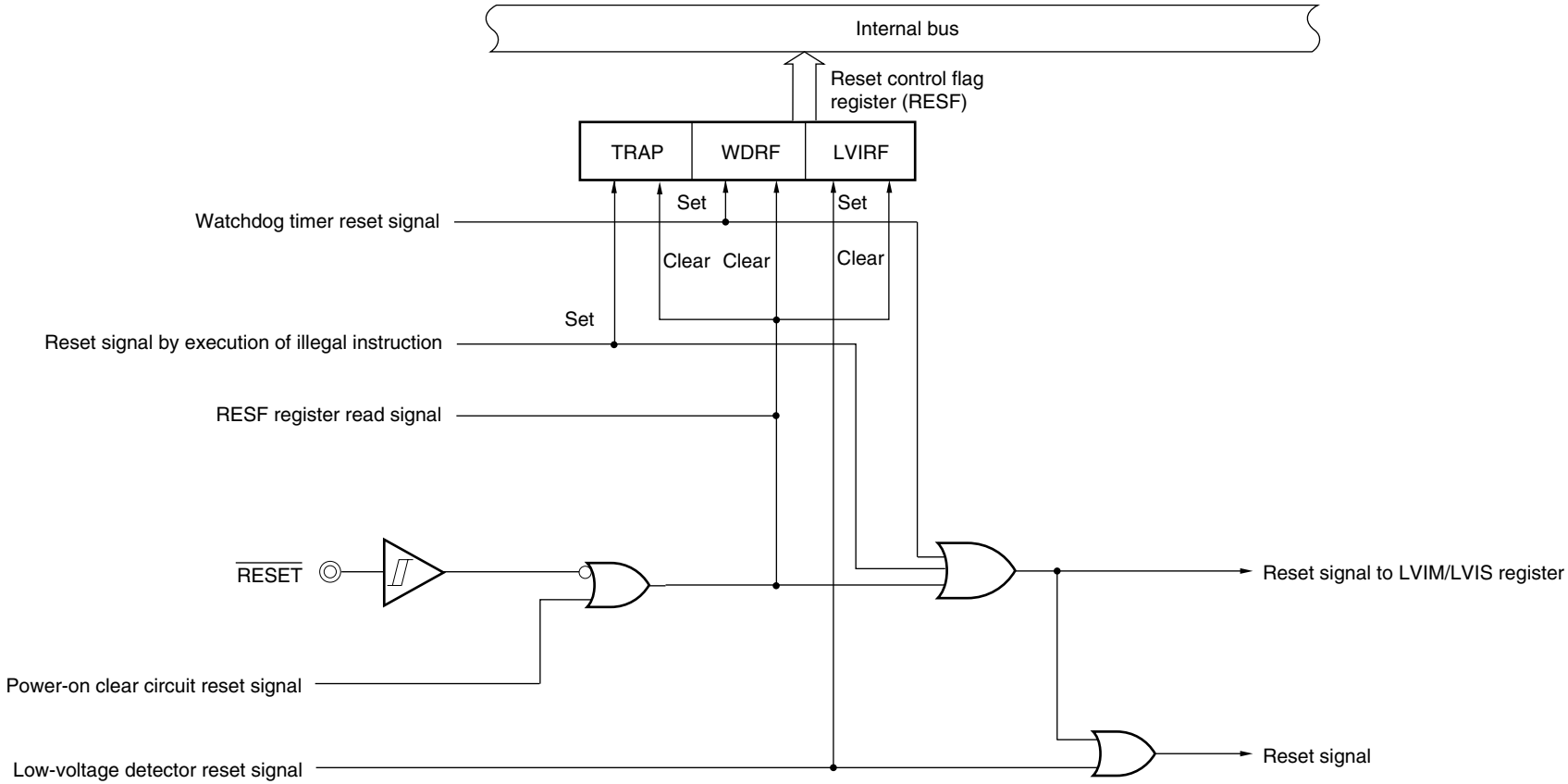
A reset is effected when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POC and LVI circuit voltage detection or execution of illegal instruction[Note], and each item of hardware is set to the status shown in Tables 18-1 and 18-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release, except for P130, which is low-level output.

When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the internal high-speed oscillation clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the internal high-speed oscillation clock (see **Figures 18-2** to **18-4**) after reset processing. Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} \geq V_{POC}$ or $V_{DD} \geq V_{LVI}$ after the reset, and program execution starts using the internal high-speed oscillation clock (see **CHAPTER 19 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 20 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.
(If an external reset is effected upon power application, the period during which the supply voltage is outside the operating range (V_{DD} < 1.8 V) is not counted in the 10 μs. However, the low-level input may be continued before POC is released.)
2. During reset input, the X1 clock, XT1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.
3. When the STOP mode is released by a reset, the RAM contents in the STOP mode are held during reset input. However, because SFR and 2nd SFR are initialized, the port pins become high-impedance, except for P130, which is set to low-level output.

Figure 18-1. Block Diagram of Reset Function

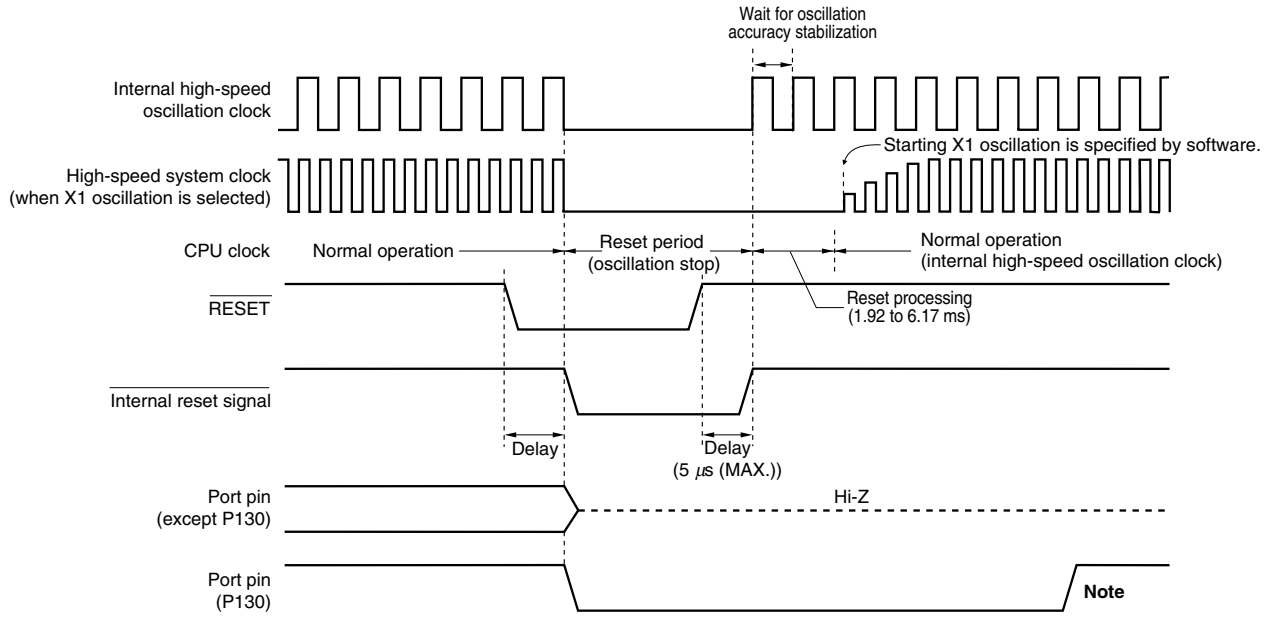


Caution An LVI circuit internal reset does not reset the LVI circuit.

Remarks 1. LVIM: Low-voltage detection register

2. LVIS: Low-voltage detection level select register

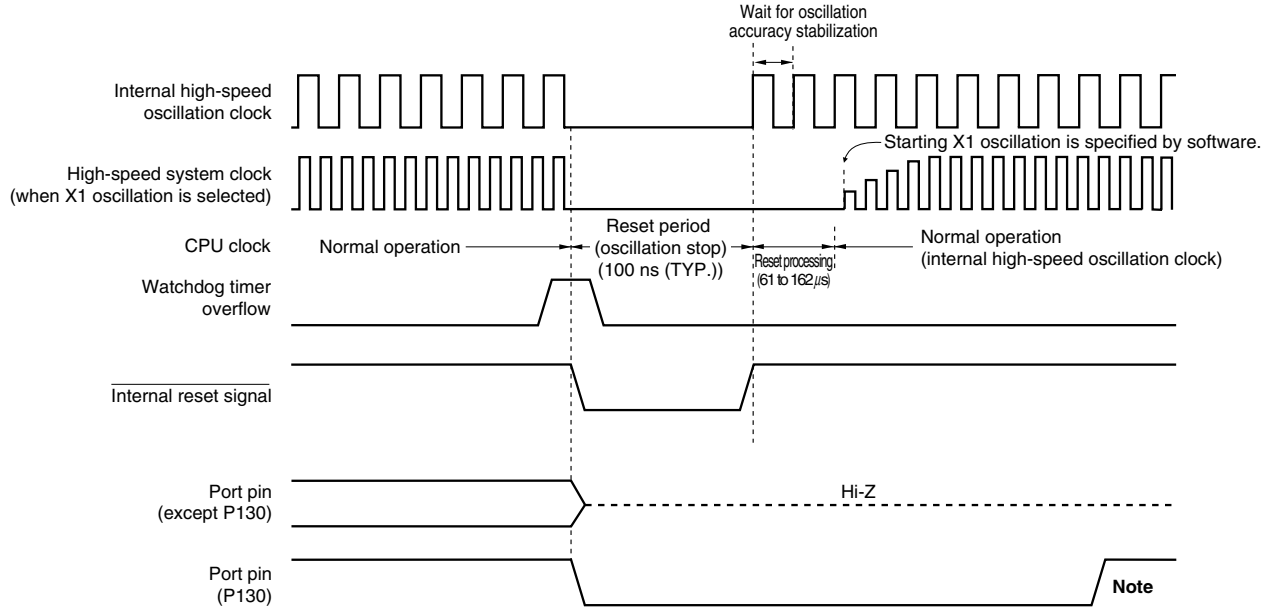
Figure 18-2. Timing of Reset by $\overline{\text{RESET}}$ Input



Note Set P130 to high-level output by software.

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

Figure 18-3. Timing of Reset Due to Watchdog Timer Overflow

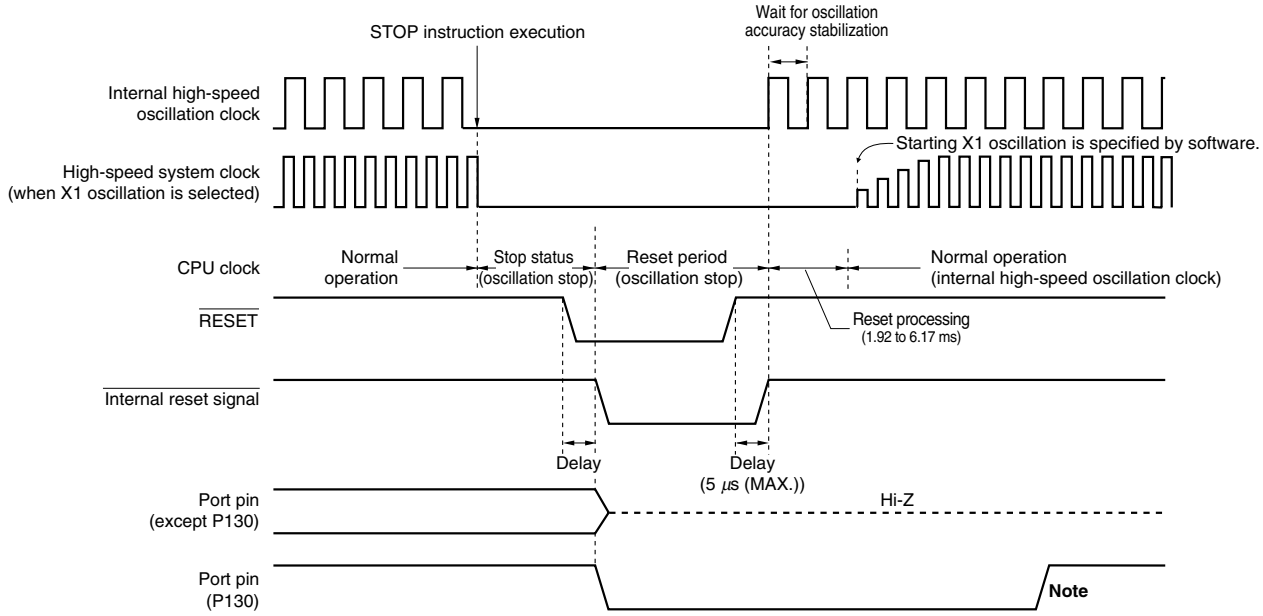


Note Set P130 to high-level output by software.

Caution A watchdog timer internal reset resets the watchdog timer.

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

Figure 18-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Note Set P130 to high-level output by software.

Remarks 1. When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.

2. For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 19 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 20 LOW-VOLTAGE DETECTOR**.

Table 18-1. Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_X	Operation stopped (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			Operable in low-current consumption mode
RAM			Operation stopped
Port (latch)			Operation stopped
Timer array unit (TAU)			
Real-time counter (RTC)			
Watch timer			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Serial array unit (SAU)			
Multiplier			
DMA controller			
Power-on-clear function			Operable
Low-voltage detection function			Operation stopped (however, operation continues at LVI reset)
External interrupt			Operation stopped

Remark f_{IH}: 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 18-2. Hardware Statuses After Reset Acknowledgment (1/3)

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]
Port registers (P0 to P7, P12 to P14) (output latches)		00H
Port mode registers (PM0 to PM7, PM12, PM14)		FFH
Port input mode registers 0 (PIM0)		00H
Port output mode registers 0 (POM0)		00H
Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU7, PU12, PU14)		00H
Clock operation mode control register (CMC)		00H
Clock operation status control register (CSC)		C0H
Processor mode control register (PMC)		00H
System clock control register (CKC)		09H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		07H
Noise filter enable registers 0, 1 (NFEN0, NFEN1)		00H
Peripheral enable registers 0 (PER0)		00H
Internal high-speed oscillator trimming register (HIOTRM)		10H
Operation speed mode control register (OSMC)		00H
Timer array unit (TAU)	Timer data registers 00, 01, 02, 03, 04, 05, 06, 07 (TDR00, TDR01, TDR02, TDR03, TDR04, TDR05, TDR06, TDR07)	0000H
	Timer mode registers 00, 01, 02, 03, 04, 05, 06, 07 (TMR00, TMR01, TMR02, TMR03, TMR04, TMR05, TMR06, TMR07)	0000H
	Timer status registers 00, 01, 02, 03, 04, 05, 06, 07 (TSR00, TSR01, TSR02, TSR03, TSR04, TSR05, TSR06, TSR07)	0000H
	Timer input select register 0 (TIS0)	00H
	Timer channel counter registers 00, 01, 02, 03, 04, 05, 06, 07 (TCR00, TCR01, TCR02, TCR03, TCR04, TCR05, TCR06, TCR07)	FFFFH
	Timer channel enable status register 0 (TE0)	0000H
	Timer channel start trigger register 0 (TS0)	0000H
	Timer channel stop trigger register 0 (TT0)	0000H
	Timer clock select register 0 (TPS0)	0000H
	Timer channel output register 0 (TO0)	0000H
	Timer channel output enable register 0 (TOE0)	0000H
	Timer channel output level register 0 (TOL0)	0000H
	Timer channel output mode register 0 (TOM0)	0000H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

Table 18-2. Hardware Statuses After Reset Acknowledgment (2/3)

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
	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
	Real-time counter control register 0 (RTCC0)	00H
	Real-time counter control register 1 (RTCC1)	00H
	Real-time counter control register 2 (RTCC2)	00H
Clock output/buzzer output controller	Clock output select registers 0, 1 (CKS0, CKS1)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[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, 12, 13 (SDR00, SDR01, SDR02, SDR03, SDR12, SDR13)	0000H
	Serial status registers 00, 01, 02, 03, 12, 13 (SSR00, SSR01, SSR02, SSR03, SSR12, SSR13)	0000H
	Serial flag clear trigger registers 00, 01, 02, 03, 12, 13 (SIR00, SIR01, SIR02, SIR03, SIR12, SIR13)	0000H
	Serial mode registers 00, 01, 02, 03, 12, 13 (SMR00, SMR01, SMR02, SMR03, SMR12, SMR13)	0020H
	Serial communication operation setting registers 00, 01, 02, 03, 12, 13 (SCR00, SCR01, SCR02, SCR03, SCR12, SCR13)	0087H
	Serial channel enable status registers 0, 1 (SE0, SE1)	0000H
	Serial channel start trigger registers 0, 1 (SS0, SS1)	0000H
	Serial channel stop trigger 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
	Input switch control register (ISC)	00H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

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

Table 18-2. Hardware Statuses After Reset Acknowledgment (3/3)

Hardware		Status After Reset Acknowledgment[Note 1]
Serial interface IIC0	Shift register 0 (IIC0)	00H
	Control register 0 (IICC0)	00H
	Slave address register 0 (SVA0)	00H
	Clock select register 0 (IICCL0)	00H
	Function expansion register 0 (IICX0)	00H
	Status register 0 (IICS0)	00H
	Flag register 0 (IICF0)	00H
Multiplier	Multiplication input data register A (MULA)	0000H
	Multiplication input data register B (MULB)	0000H
	Higher multiplication result storage register (MULOH)	0000H
	Lower multiplication result storage register (MULOL)	0000H
Key interrupt	Key return mode register (KRM)	00H
Reset function	Reset control flag register (RESF)	00H[Note 2]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note 3]
	Low-voltage detection level select register (LVIS)	0EH[Note 2]
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 (EGP0, EGP1)	00H
	External interrupt falling edge enable registers (EGN0, EGN1)	00H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. These values vary depending on the reset source.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by LVI
RESF	TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held
	WDRF bit			Held	Set (1)	Held
	LVIRF bit			Held	Held	Set (1)
LVIS		Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Held

3. This value varies depending on the reset source and the option byte.

18.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0R/KE3. 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 18-5. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: 00H[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 is cleared.
1	Internal reset request is generated.

WDRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by low-voltage detector (LVI)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

Notes 1. The value after reset varies depending on the reset source.

2. 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 18-3.

Table 18-3. 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 LVI
TRAP	Cleared (0)	Cleared (0)	Set (1)	Held	Held
WDRF			Held	Set (1)	Held
LVIRF			Held	Held	Set (1)

CHAPTER 19 POWER-ON-CLEAR CIRCUIT

19.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage (V_{DD}) exceeds 1.59 V ±0.09 V[Note].

  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[Note].

- Compares supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.09 V[Note]), generates internal reset signal when $V_{DD} < V_{POC}$.

  Note These are preliminary values and subject to change.

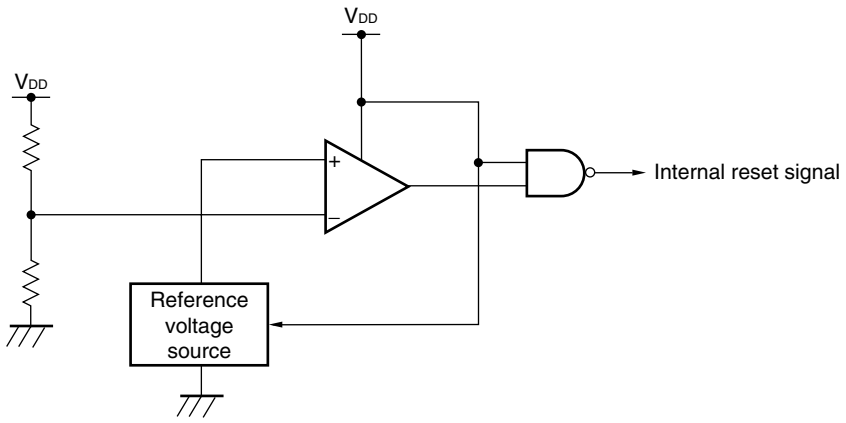
  Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

  Remark This product incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), low-voltage-detector (LVI), or illegal instruction execution. RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT or LVI.
  For details of RESF, see **CHAPTER 18 RESET FUNCTION**.

19.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 19-1.

Figure 19-1. Block Diagram of Power-on-Clear Circuit



19.3 Operation of Power-on-Clear Circuit

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{POC} = 1.59 V ±0.09 V[Note]), 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[Note].

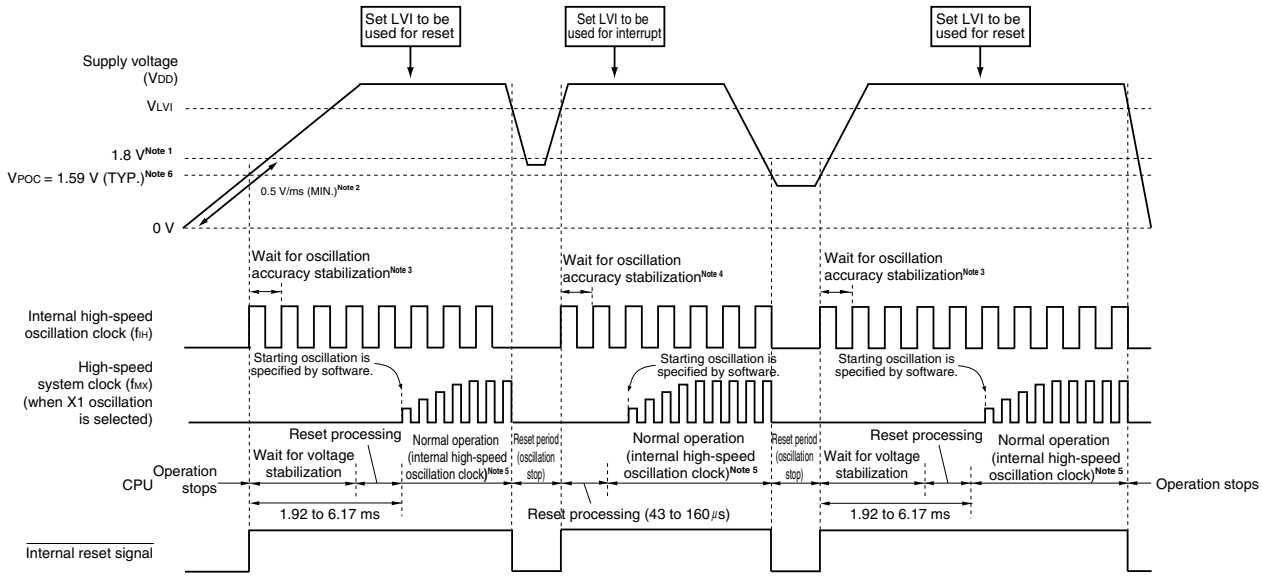
- The supply voltage (V_{DD}) and detection voltage (V_{POC} = 1.59 V ±0.09 V[Note]) are compared. When $V_{DD} < V_{POC}$, the internal reset signal is generated.

  Note These are preliminary values and subject to change.

The timing of generation of the internal reset signal by the power-on-clear circuit and low-voltage detector is shown below.

Figure 19-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)

(1) When LVI is OFF upon power application (option byte: LVIOFF = 1)



Notes 1. The operation guaranteed range is 1.8 V ≤ V_{DD} ≤ 5.5 V. To make the state at lower than 1.8 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. If the rate at which the voltage rises to 1.8 V after power application is slower than 0.5 V/ms (MIN.), input a low level to the $\overline{\text{RESET}}$ pin before the voltage reaches to 1.8 V, or set LVI to ON by default by using an option byte (option byte: LVIOFF = 0).

3. The internal voltage stabilization time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

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

5. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

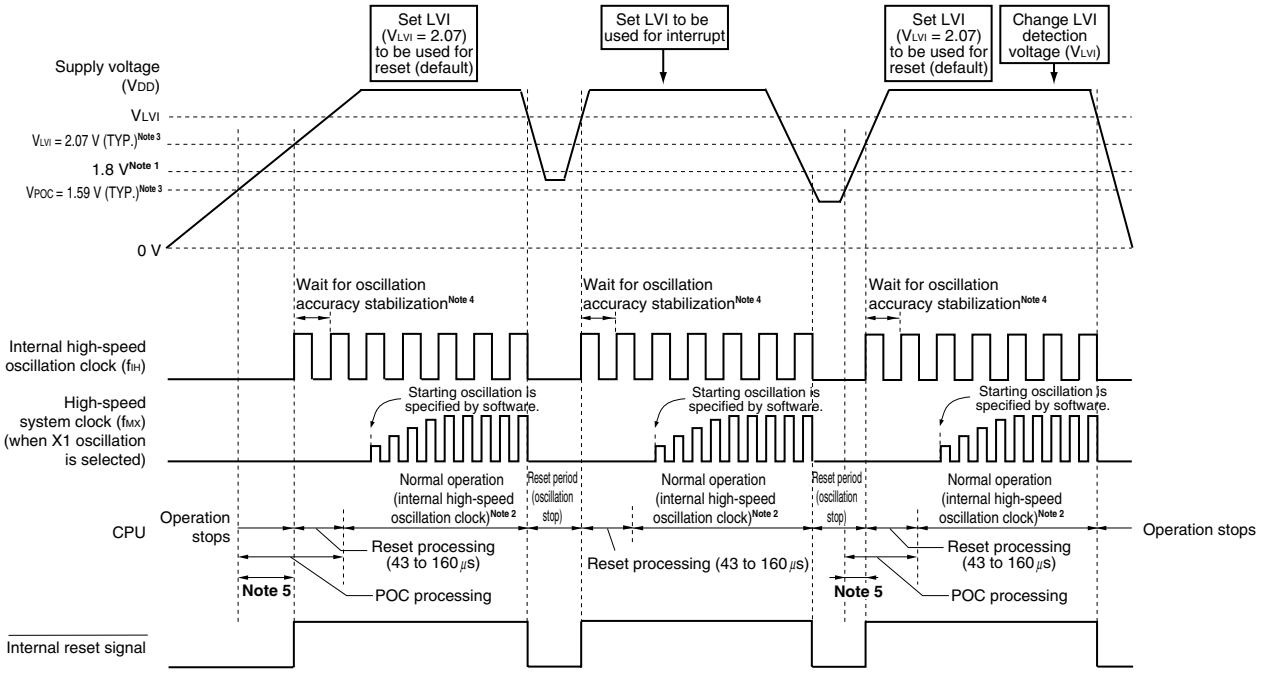
6. This is a preliminary value and subject to change.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 20 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POC}: POC detection voltage

Figure 19-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)

(2) When LVI is ON upon power application (option byte: LVIOFF = 0)



Notes 1. The operation guaranteed range is 1.8 V ≤ V_{DD} ≤ 5.5 V. To make the state at lower than 1.8 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

3. These are preliminary values and subject to change.

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

5. The following times are required between reaching the POC detection voltage (1.59 V (TYP.)) and starting normal operation.

- When the time to reach 2.07 V (TYP.) from 1.59 V (TYP.) is less than 6.17 ms:
  A POC processing time of 1.92 to 6.33 ms is required between reaching 1.59 V (TYP.) and starting normal operation.
- When the time to reach 2.07 V (TYP.) from 1.59 V (TYP.) is greater than 6.17 ms:
  A reset processing time of 43 to 160 μs is required between reaching 2.07 V (TYP.) and starting normal operation.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 20 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POC}: POC detection voltage

19.4 Cautions for Power-on-Clear Circuit

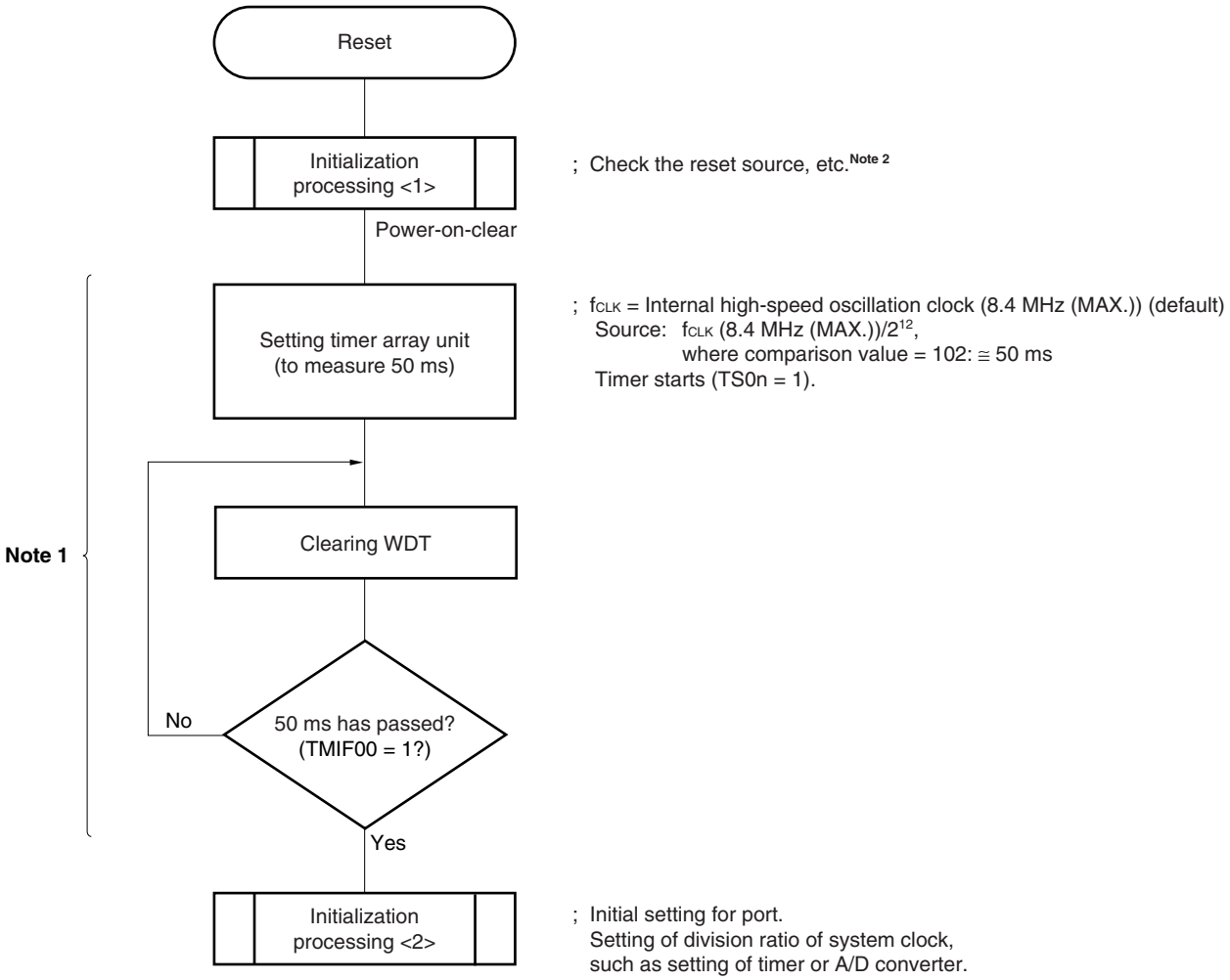
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 19-3. Example of Software Processing After Reset Release (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage



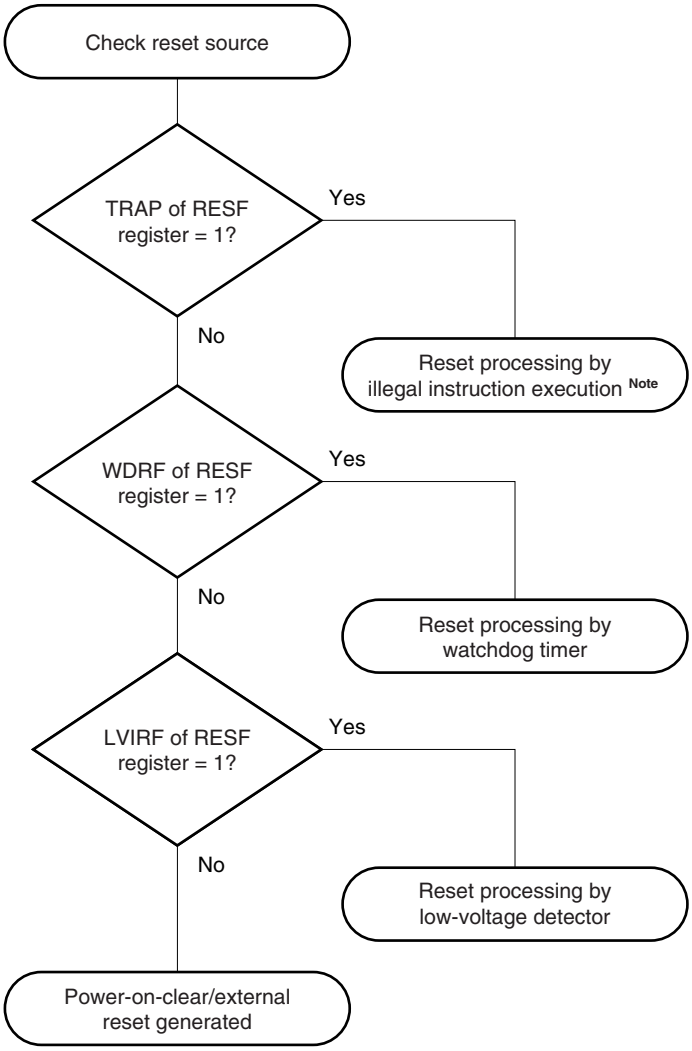
Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Remark n: Channel number (n = 0 to 7)

Figure 19-3. 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 20 LOW-VOLTAGE DETECTOR

20.1 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[Note]), 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 or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.2 V[Note]). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 V[Note]).
- The supply voltage (V_{DD}) or the input voltage from the external input pin (EXLVI) can be selected to be detected by software.
- A reset or an interrupt can be selected to be generated after detection by software.
- Detection levels (V_{LVI},16 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

Note This is a preliminary value and subject to change.

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

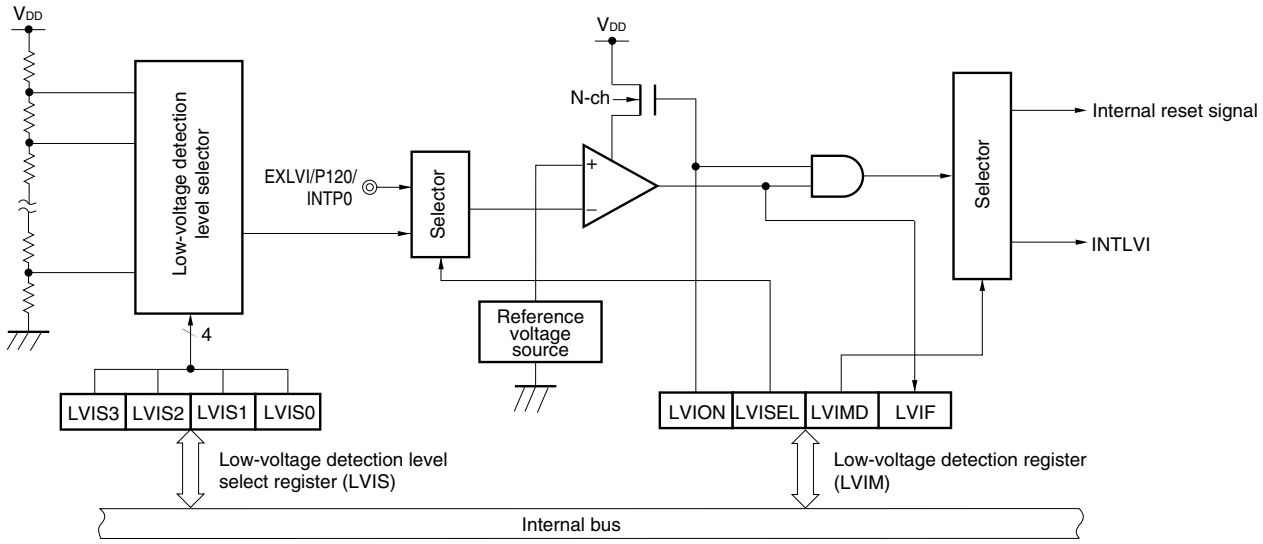
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

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 18 RESET FUNCTION**.

20.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 20-1.

Figure 20-1. Block Diagram of Low-Voltage Detector



20.3 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 20-2. 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.

Notes 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.))
- Detection delay time (200 μs (MAX.))

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 1. To stop LVI, follow either of the procedures below.

- **When using 8-bit memory manipulation instruction: Write 00H to LVIM.**
- **When using 1-bit memory manipulation instruction: Clear LVION to 0.**

2. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

3. When LVI is used in interrupt mode (LVIMD = 0) and LVISEL is set to 0, an interrupt request signal (INTLVI) that disables LVI operation (clears LVION) when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}) (if LVISEL = 1, input voltage of external input pin (EXLVI) is less than or equal to the detection voltage (V_{EXLVI})) is generated and LVIIF may be set to 1.

(2) Low-voltage detection level select register (LVIS)

This register selects the low-voltage detection level.

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

Reset signal generation input sets this register to 0EH.

Figure 20-3. Format of Low-Voltage Detection Level Select Register (LVIS)

Address: FFFAAH After reset: 0EH[Note 1] 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 2]
0	0	0	1	V_{LVI1} (4.07 ±0.1 V)[Note 2]
0	0	1	0	V_{LVI2} (3.92 ±0.1 V)[Note 2]
0	0	1	1	V_{LVI3} (3.76 ±0.1 V)[Note 2]
0	1	0	0	V_{LVI4} (3.61 ±0.1 V)[Note 2]
0	1	0	1	V_{LVI5} (3.45 ±0.1 V)[Note 2]
0	1	1	0	V_{LVI6} (3.30 ±0.1 V)[Note 2]
0	1	1	1	V_{LVI7} (3.15 ±0.1 V)[Note 2]
1	0	0	0	V_{LVI8} (2.99 ±0.1 V)[Note 2]
1	0	0	1	V_{LVI9} (2.84 ±0.1 V)[Note 2]
1	0	1	0	V_{LVI10} (2.68 ±0.1 V)[Note 2]
1	0	1	1	V_{LVI11} (2.53 ±0.1 V)[Note 2]
1	1	0	0	V_{LVI12} (2.38 ±0.1 V)[Note 2]
1	1	0	1	V_{LVI13} (2.22 ±0.1 V)[Note 2]
1	1	1	0	V_{LVI14} (2.07 ±0.1 V)[Note 2]
1	1	1	1	V_{LVI15} (1.91 ±0.1 V)[Note 2]

Notes 1. 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.

2. These are preliminary values and subject to change.

Cautions 1. Be sure to clear bits 4 to 7 to “0”.

Cautions 2. Change the LVIS value with either of the following methods.

- **When changing the value after stopping LVI**
  - **<1> Stop LVI (LVION = 0).**
  - **<2> Change the LVIS register.**
  - **<3> Set to the mode used as an interrupt (LVIMD = 0).**
  - **<4> Mask LVI interrupts (LVIMK = 1).**
  - **<5> Enable LVI operation (LVION = 1).**
  - **<6> Before cancelling the LVI interrupt mask (LVIMK = 0), clear it with software because an LVIIF flag may be set when LVI operation is enabled.**
- **When changing the value after setting to the mode used as an interrupt (LVIMD = 0)**
  - **<1> Mask LVI interrupts (LVIMK = 1).**
  - **<2> Set to the mode used as an interrupt (LVIMD = 0).**
  - **<3> Change the LVIS register.**
  - **<4> Before cancelling the LVI interrupt mask (LVIMK = 0), clear it with software because an LVIIF flag may be set when the LVIS register is changed.**

3. When an input voltage from the external input pin (EXLVI) is detected, the detection voltage (V_{EXLVI}) is fixed. Therefore, setting of LVIS is not necessary.

(3) Port mode register 12 (PM12)

When using the P120/EXLVI/INTP0 pin for external low-voltage detection potential input, set PM120 to 1. At this time, the output latch of P120 may be 0 or 1.

PM12 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 20-4. 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)

20.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

(1) Used as reset (LVIMD = 1)

- If LVISEL = 0, compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), generates an internal reset signal when $V_{DD} < V_{LVI}$, and releases internal reset when $V_{DD} \geq V_{LVI}$.
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage (V_{EXLVI}), generates an internal reset signal when EXLVI < V_{EXLVI}, and releases internal reset when EXLVI ≥ V_{EXLVI}.

Remark The low-voltage detector (LVI) can be set to ON by an option byte by default. If it is set to ON to raise the power supply from the POC detection voltage or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.2 V[Note]). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 V[Note]).

(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[Note]). 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).

Note This is a preliminary value and subject to change.

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

20.4.1 When used as reset

(1) When detecting level of supply voltage (V_{DD})

(a) When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1)

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD})) (default value).

<3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).

<4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<5> Use software to wait for the following periods of time (Total 410 μs).

- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))
- Detection delay time (200 μs (MAX.))

<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 20-5 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

Cautions 1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <4>.

2. If supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}) when LVIMD is set to 1, an internal reset signal is not generated.

- When stopping operation

Either of the following procedures must be executed.

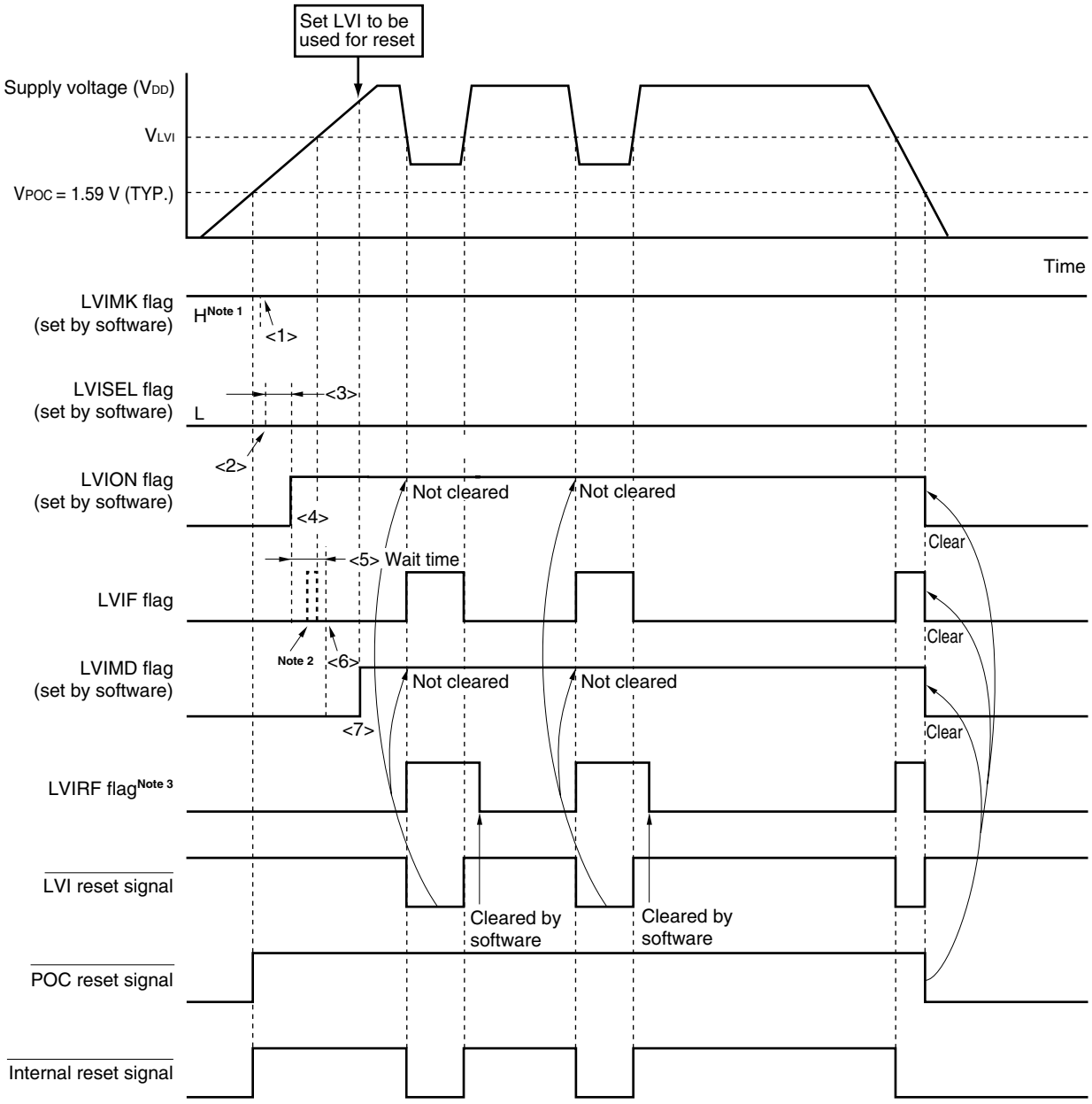
- When using 8-bit memory manipulation instruction:

Write 00H to LVIM.

- When using 1-bit memory manipulation instruction:

Clear LVIMD to 0 and then LVION to 0.

Figure 20-5. 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 18 RESET FUNCTION**.

Remark <1> to <7> in Figure 20-5 above correspond to <1> to <7> in the description of “When starting operation” in **20.4.1 (1) (a) When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1).**

(b) When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0)

- When starting operation

  Start in the following initial setting state.

  - Set bit 7 (LVION) of LVIM to 1 (enables LVI operation)
  - Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD}))
  - Set the low-voltage detection level selection register (LVIS) to 0EH (default value: V_{LVI} = 2.07 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 20-6 shows the timing of the internal reset signal generated by the low-voltage detector.

- When stopping operation

  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:

    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:

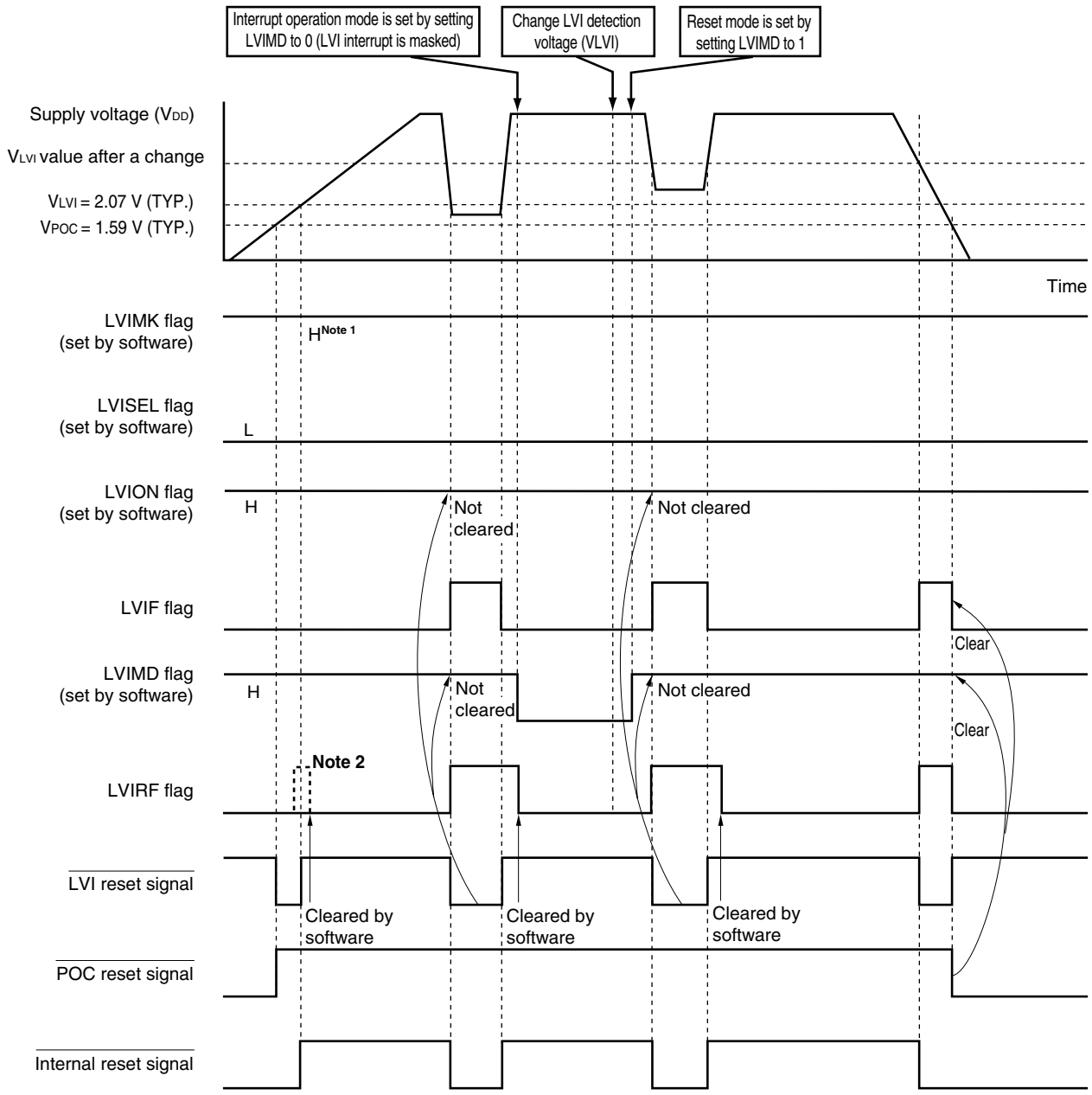
    Clear LVIMD to 0 and then LVION to 0.

Caution Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- **Does not perform low-voltage detection during LVION = 0.**
- **If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution.**

  This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.

Figure 20-6. 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 18 RESET FUNCTION**.

(2) When detecting level of input voltage from external input pin (EXLVI)

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).

<3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<4> Use software to wait for the following periods of time (Total 410 μs).

- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))
- Detection delay time (200 μs (MAX.))

<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 20-7 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <6> above.

Cautions 1. <1> must always be executed. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.

2. If input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.

3. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

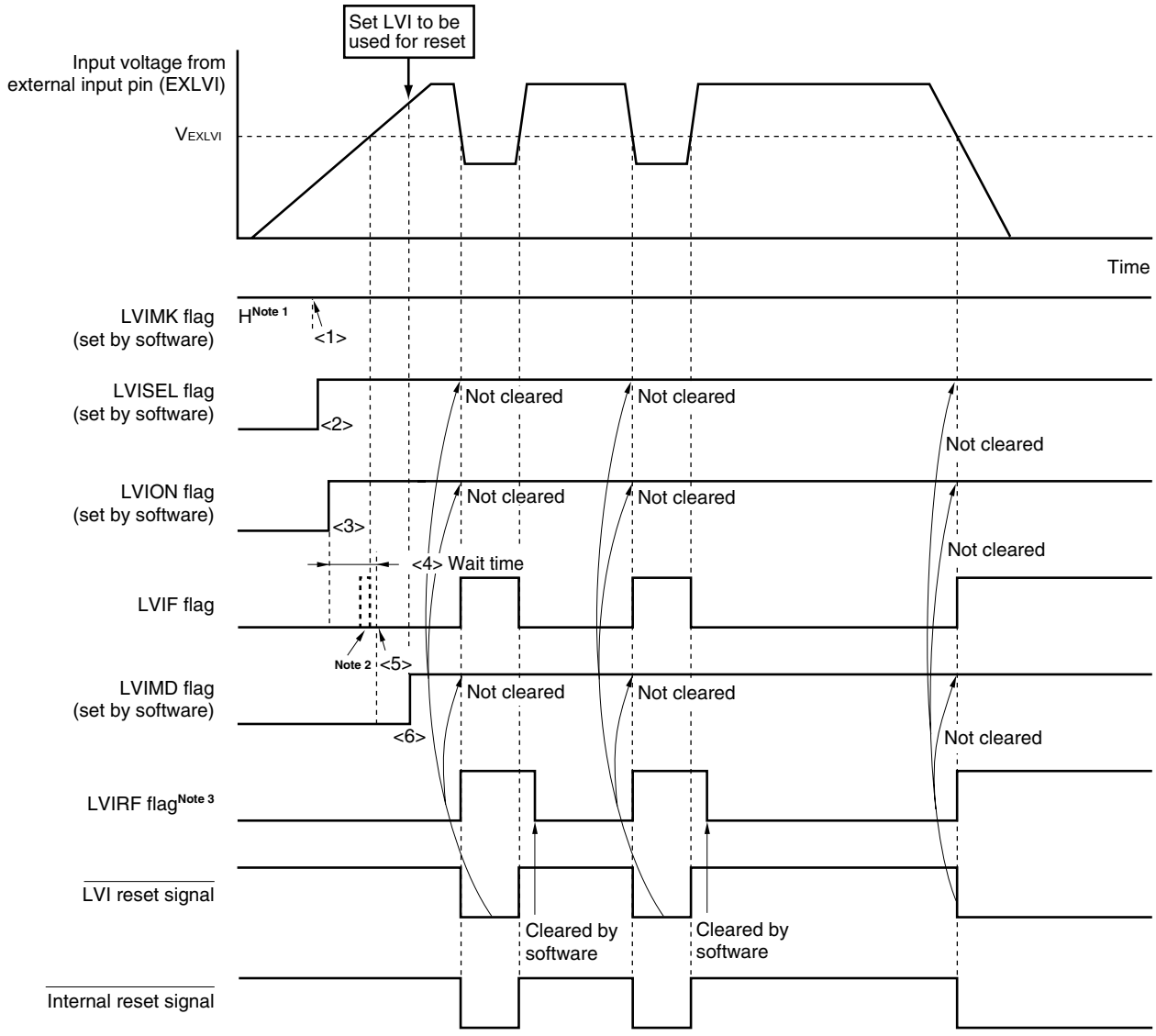
- When stopping operation

Either of the following procedures must be executed.

- When using 8-bit memory manipulation instruction:
  Write 00H to LVIM.

- When using 1-bit memory manipulation instruction:
  Clear LVIMD to 0 and then LVION to 0.

Figure 20-7. 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 18 RESET FUNCTION**.

Remark <1> to <6> in Figure 20-7 above correspond to <1> to <6> in the description of **"**When starting operation" in **20.4.1 (2) When detecting level of input voltage from external input pin (EXLVI)**.

20.4.2 When used as interrupt

(1) When detecting level of supply voltage (V_{DD})

(a) When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1)

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD})) (default value).
Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).

<3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).

<4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<5> Use software to wait for the following periods of time (Total 410 μs).
- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))
- Detection delay time (200 μs (MAX.))

<6> Confirm that "supply voltage (V_{DD}) $\geq$ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, at bit 0 (LVIF) of LVIM.

<7> Clear the interrupt request flag of LVI (LVIIF) to 0.

<8> Release the interrupt mask flag of LVI (LVIMK).

<9> Execute the EI instruction (when vector interrupts are used).

Figure 20-8 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <8> above.

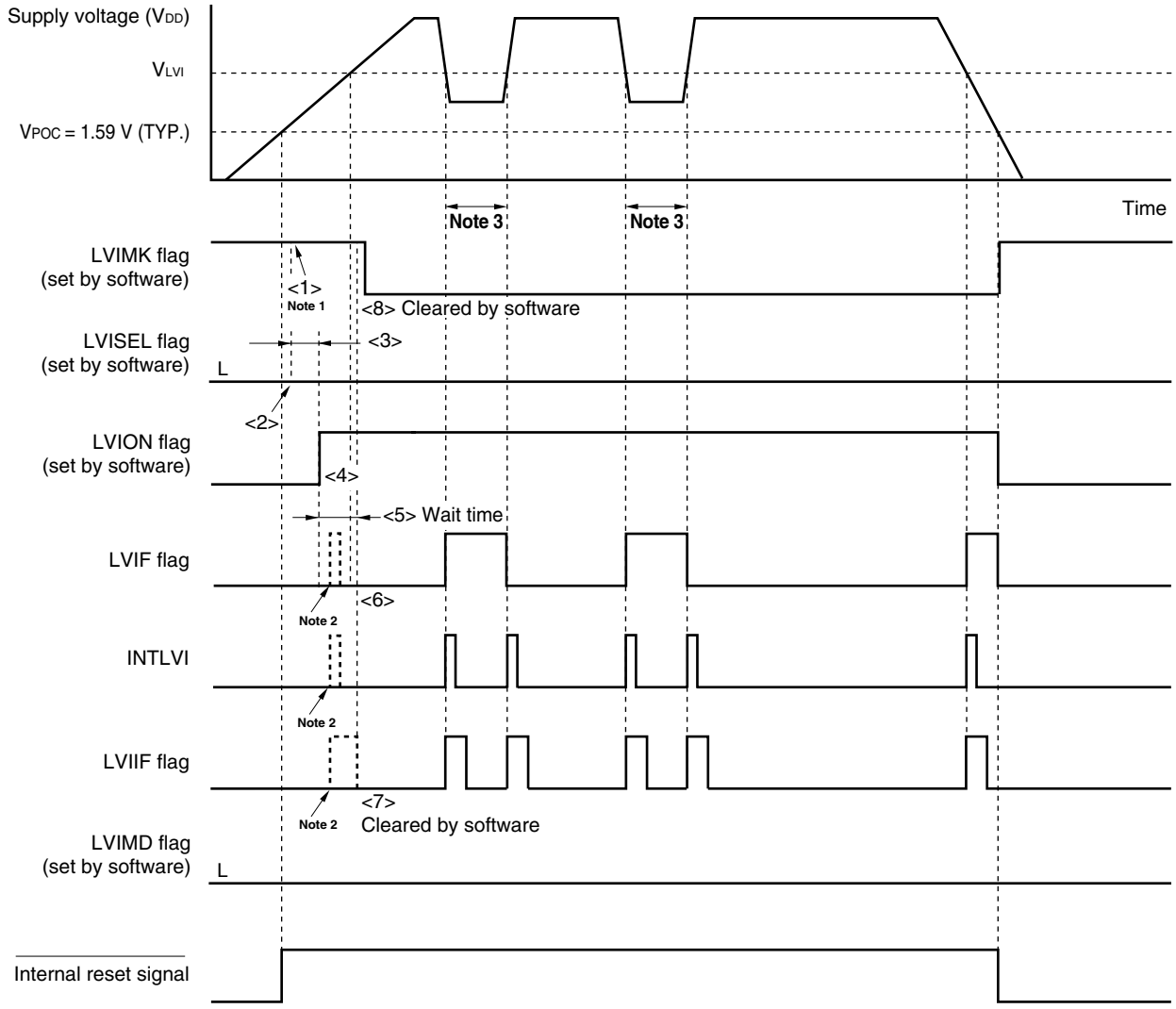
- When stopping operation

Either of the following procedures must be executed.

- When using 8-bit memory manipulation instruction:
  Write 00H to LVIM.

- When using 1-bit memory manipulation instruction:
  Clear LVION to 0.

Figure 20-8. 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.

Remark <1> to <8> in Figure 20-8 above correspond to <1> to <8> in the description of "When starting operation" in **20.4.2 (1) (a) When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1).**

(b) When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0)

- When starting operation

<1> Start in the following initial setting state.

- Set bit 7 (LVION) of LVIM to 1 (enables LVI operation)
- Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD}))
- Set the low-voltage detection level selection register (LVIS) to 0EH (default value: V_{LVI} = 2.07 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 "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 20-9 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <3> above.

- When stopping operation

Either of the following procedures must be executed.

- When using 8-bit memory manipulation instruction:

Write 00H to LVIM.

- When using 1-bit memory manipulation instruction:

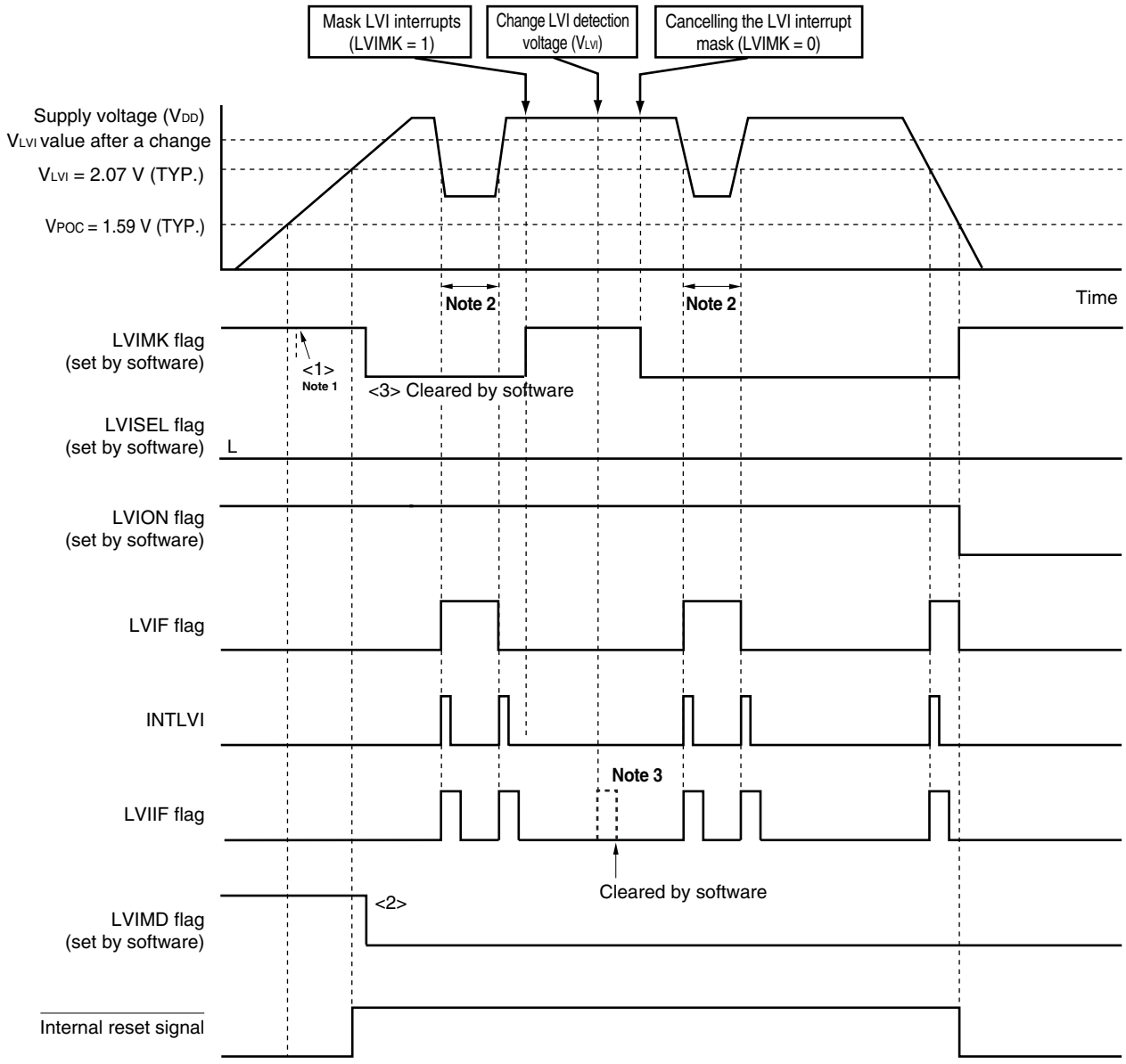
Clear LVION to 0.

Cautions 1. Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- **Does not perform low-voltage detection during LVION = 0.**
- **If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution. This is due to the fact that while the pulse width detected by LVI must be 200 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 18 RESET FUNCTION.

Figure 20-9. 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.

Remark <1> to <3> in Figure 20-9 above correspond to <1> to <3> in the description of "When starting operation" in **20.4.2 (1) (b) When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0).**

(2) When detecting level of input voltage from external input pin (EXLVI)

- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
    Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <4> Use software to wait for the following periods of time (Total 410 μs).
    - Operation stabilization time (10 μs (MAX.))
    - Minimum pulse width (200 μs (MIN.))
    - Detection delay time (200 μs (MAX.))
  - <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 20-10 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

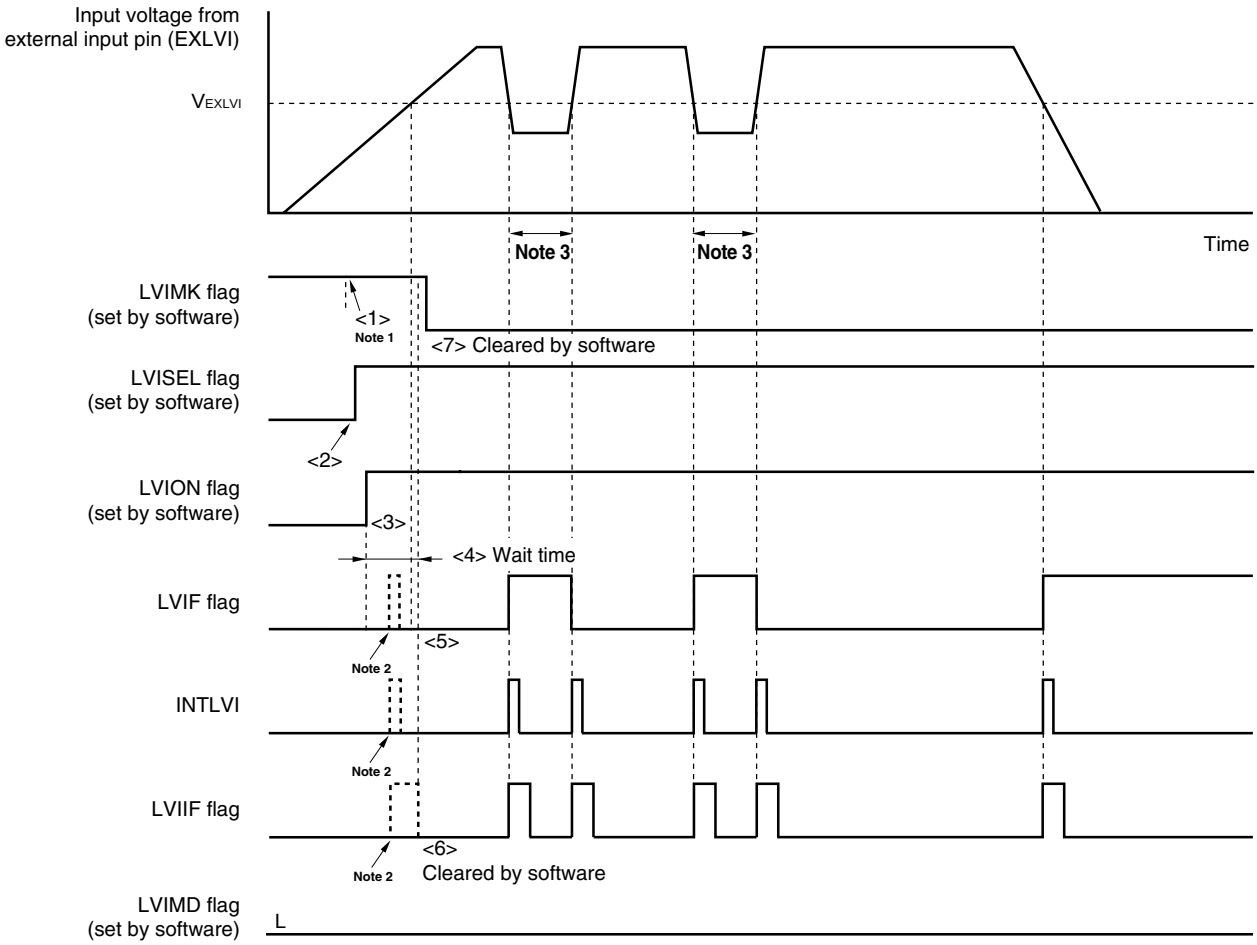
  Caution Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation
  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.

  - When using 1-bit memory manipulation instruction:
    Clear LVION to 0.

Figure 20-10. 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 20-10 above correspond to <1> to <7> in the description of "When starting operation" in **20.4.2 (2) When detecting level of input voltage from external input pin (EXLVI)**.

20.5 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>

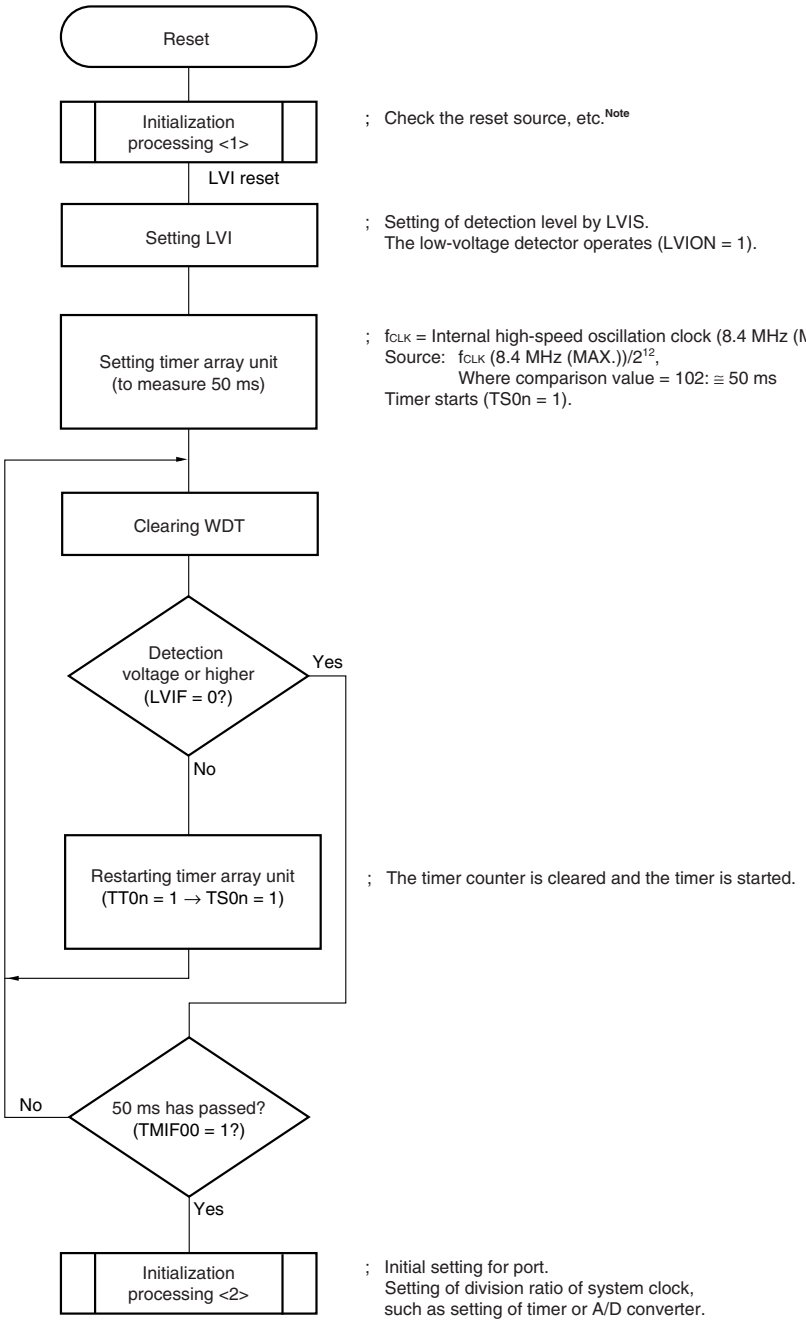
After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (see **Figure 20-11**).

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 20-11. Example of Software Processing After Reset Release (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



Note A flowchart is shown on the next page.

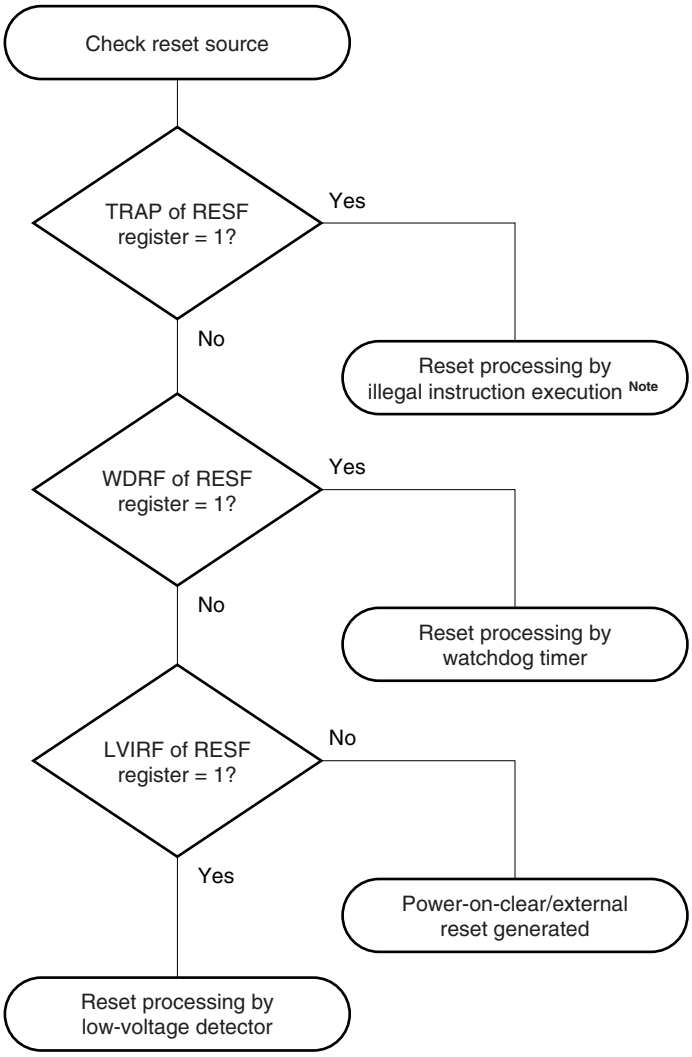
Remarks 1. n: Channel number (n = 0 to 7)

2. If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to “1”, the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

Figure 20-11. 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.

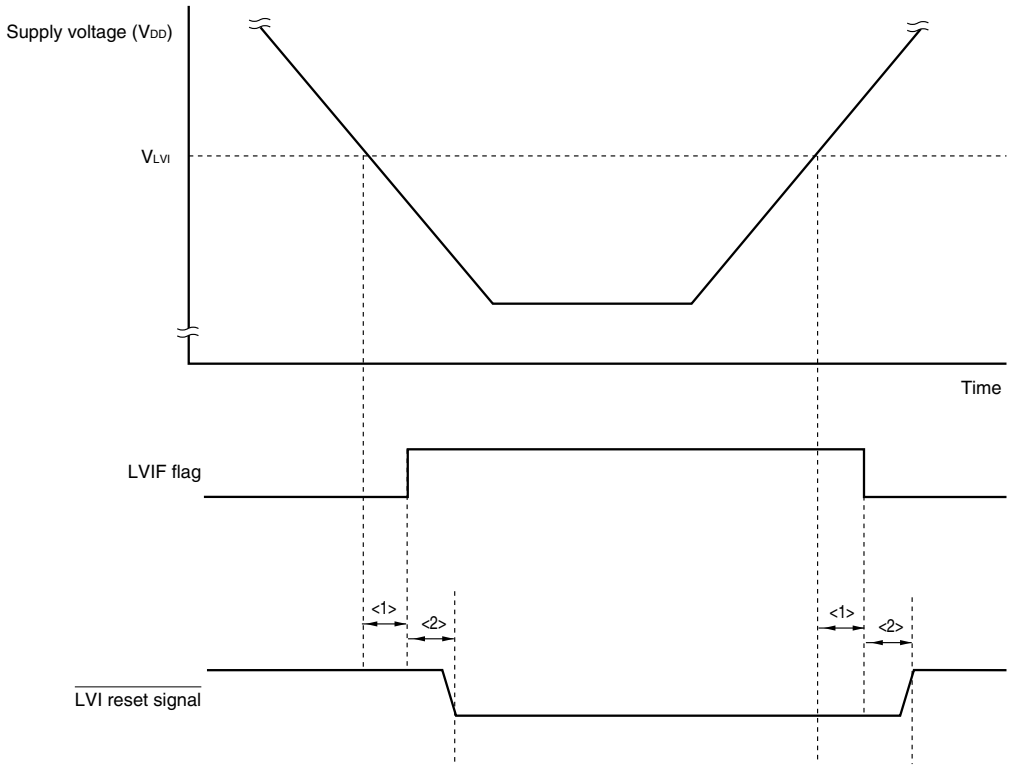
- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

(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 20-12**).

Figure 20-12. 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.))

<2> : Detection delay time (200 μs (MAX.))

CHAPTER 21 REGULATOR

21.1 Regulator Overview

The 78K0R/KE3 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 2.5 V (typ.), and in the low consumption current mode, 1.8 V (typ.).

21.2 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 21-1. Format of Regulator Mode Control Register (RMC)

Address: F00F4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMC								

RMC[7:0]	Control of output voltage of regulator
5AH	Fixed to low consumption current mode (1.8 V)
00H	Switches normal current mode (2.5 V) and low consumption current mode (1.8 V) according to the condition (refer to **Table 21-1**)
Other than above	Setting prohibited

Cautions 1. The RMC register can be rewritten only in the low consumption current mode (refer to Table 21-1). In other words, rewrite this register during CPU operation with the subsystem clock (f_{XT}) while the high-speed system clock (f_{MX}) and high-speed internal oscillation clock (f_{IH}) are both stopped.

2. When using the setting fixed to the low consumption current mode, the RMC register can be used in the following cases.

<When X1 clock is selected as the CPU clock> $f_X \leq 5$ MHz and $f_{CLK} \leq 5$ MHz

<When the high-speed internal oscillation clock, external input clock, or subsystem clock are selected for the CPU clock> $f_{CLK} \leq 5$ MHz

3. The self-programming function is disabled in the low consumption current mode.

Table 21-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
Low consumption current mode	1.8 V	During system reset
		In STOP mode (except during OCD mode)
		When both the high-speed system clock (f_{MX}) and the high-speed internal oscillation clock (f_{IH}) are stopped during CPU operation with the subsystem clock (f_{XT})
		When both the high-speed system clock (f_{MX}) and the high-speed internal oscillation clock (f_{IH}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{XT}) has been set
Normal current mode	2.5 V	Other than above

CHAPTER 22 OPTION BYTE

22.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the 78K0R/KE3 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).

22.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)

(1) 000C0H/010C0H

- ○ Operation of watchdog timer
  - • Operation is stopped or enabled in the HALT or STOP mode.
- ○ Setting of interval time of watchdog timer
- ○ Operation of watchdog timer
  - • Operation is stopped or enabled.
- ○ Setting of window open period of watchdog timer
- ○ Setting of interval interrupt of watchdog timer
  - • Used or not used

Caution Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

(2) 000C1H/010C1H

- ○ Setting of LVI upon reset release (upon power application)
  - • LVI is ON or OFF by default upon reset release (reset by $\overline{\text{RESET}}$ pin excluding LVI, POC, WDT, or illegal instructions).

Caution Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

(3) 000C2H/010C2H

- ○ Be sure to set FFH, as these addresses are reserved areas.

Caution Set FFH to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

22.1.2 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.

22.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 22-1. Format of User Option Byte (000C0H/010C0H) (1/2)

Address: 000C0H/010C0H[Note 1]

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINIT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period[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
0	0	0	$2^{10}/f_{IL}$ (3.88 ms)
0	0	1	$2^{11}/f_{IL}$ (7.76 ms)
0	1	0	$2^{12}/f_{IL}$ (15.52 ms)
0	1	1	$2^{13}/f_{IL}$ (31.03 ms)
1	0	0	$2^{15}/f_{IL}$ (124.12 ms)
1	0	1	$2^{17}/f_{IL}$ (496.48 ms)
1	1	0	$2^{18}/f_{IL}$ (992.97 ms)
1	1	1	$2^{20}/f_{IL}$ (3971.88 ms)

Figure 22-1. 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.

Remarks 1. f_{IL}: Internal low-speed oscillation clock frequency

2. ( ): f_{IL} = 264 kHz (MAX.)

Figure 22-2. Format of Option Byte (000C1H/010C1H)

Address: 000C1H/010C1H[Note]

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

LVIOFF	Setting of LVI on power application
0	LVI is ON by default (LVI default start function enabled) upon reset release (upon power application)
1	LVI is OFF by default (LVI default start function stopped) upon reset release (upon power application)

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Cautions 1. Be sure to set bits 7 to 1 to "1".

2. Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- **Does not perform low-voltage detection during LVION = 0.**
- **If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution.**
  This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.

Figure 22-3. 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.

22.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 22-4. Format of On-chip Debug Option Byte (000C3H/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	Erases data of flash memory in case of failures in enabling on-chip debugging and authenticating on-chip debug security ID.
1	1	Does not erases data of flash memory in case of failures in enabling on-chip debugging and 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.

22.4 Setting of Option Byte

Set the user option byte and on-chip debug option byte using a linker option of assembler package RA78K0R.
For how to set the option byte, refer to **RA78K0R Assembler Package User's Manual**.

Remark The option byte is referenced during reset processing. For the timing of reset processing, see **CHAPTER 18 RESET FUNCTION**.

CHAPTER 23 FLASH MEMORY

The 78K0R/KE3 incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board.

23.1 Writing with Flash Memory Programmer

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/KE3 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/KE3 is mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

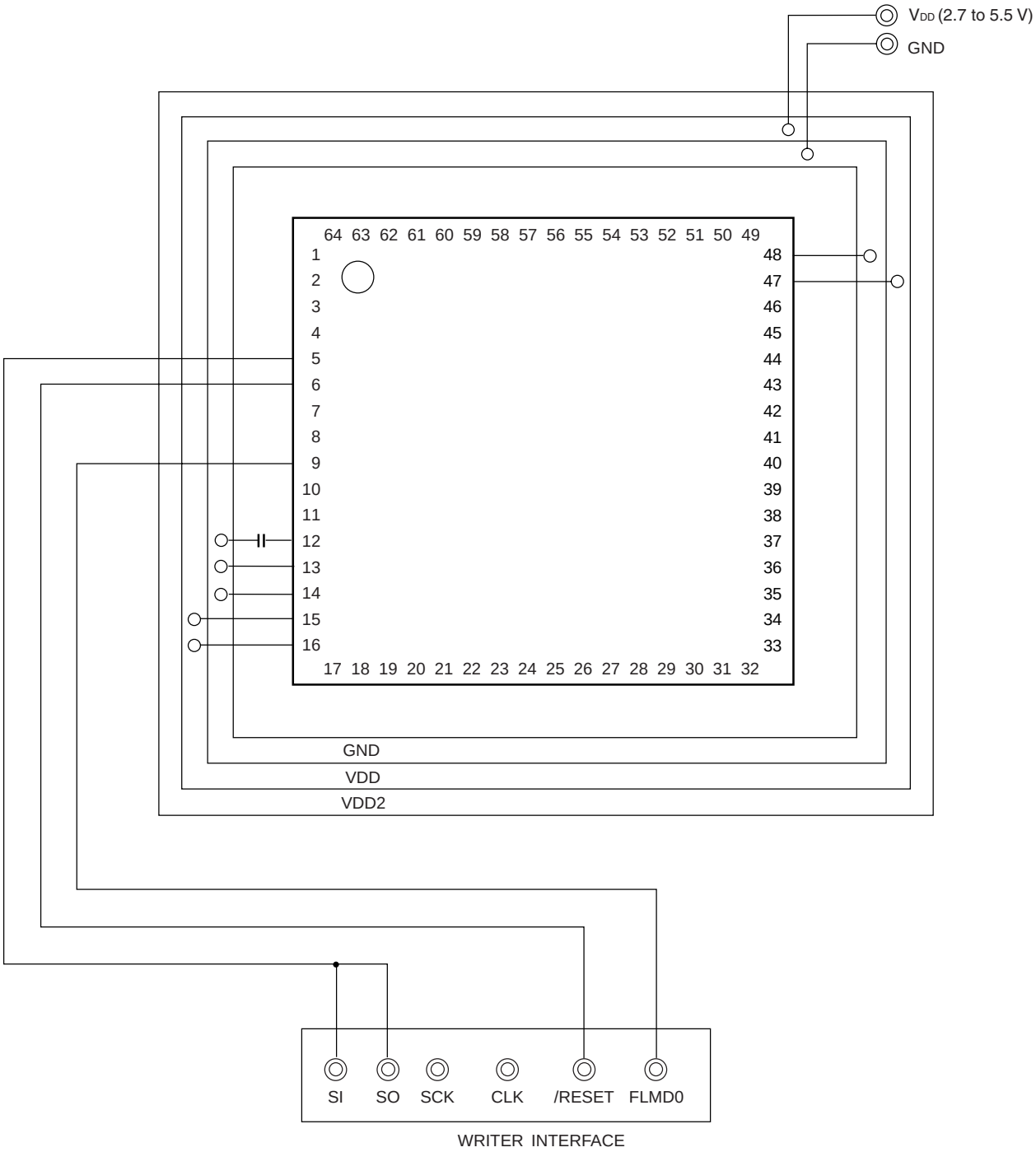
<R> **Table 23-1. Wiring Between 78K0R/KE3 and Dedicated Flash Memory Programmer**

Pin Configuration of Dedicated Flash Memory Programmer			Pin Name	Pin No.	
Signal Name	I/O	Pin Function		LQFP (12x12), LQFP (10x10), TQFP (7x7) Note	FBGA (5x5) Note
SI/RxD	Input	Receive signal	TOOL0/P40	5	D6
SO/TxD	Output	Transmit signal			
SCK	Output	Transfer clock	–	–	–
CLK	Output	Clock output	–	–	–
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	6	E7
FLMD0	Output	Mode signal	FLMD0	9	E8
V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	15	B7
			EV_{DD}	16	A8
			AV_{REF}	47	G1
GND	–	Ground	V_{SS}	13	C7
			EV_{SS}	14	B8
			AV_{SS}	48	H1

Note Under development

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

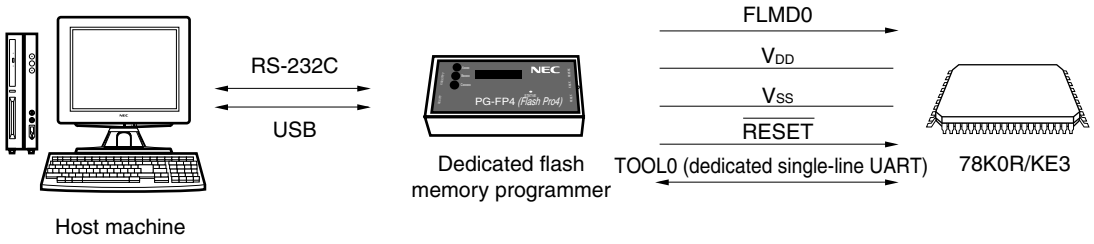
Figure 23-1. Example of Wiring Adapter for Flash Memory Writing (GF Package)



23.2 Programming Environment

The environment required for writing a program to the flash memory of the 78K0R/KE3 is illustrated below.

Figure 23-2. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

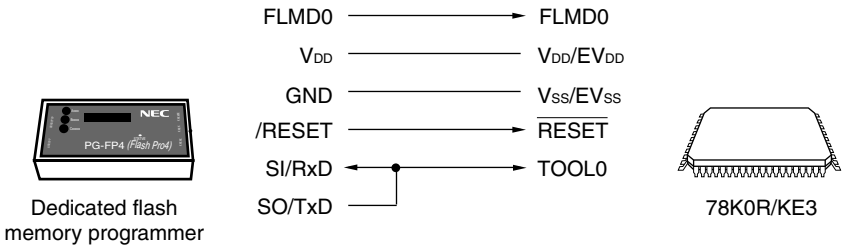
To interface between the dedicated flash memory programmer and the 78K0R/KE3, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

23.3 Communication Mode

Communication between the dedicated flash memory programmer and the 78K0R/KE3 is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the 78K0R/KE3.

Transfer rate: 115,200 bps to 1,000,000 bps

Figure 23-3. Communication with Dedicated Flash Memory Programmer



When using the FlashPro4 as the dedicated flash memory programmer, the FlashPro4 generates the following signals for the 78K0R/KE3. For details, refer to the user's manual for the FlashPro4.

Table 23-2. Pin Connection

FlashPro4			78K0R/KE3	Connection
Signal Name	I/O	Pin Function	Pin Name	
FLMD0	Output	Mode signal	FLMD0	◎
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, EV_{DD}, AV_{REF}	◎
GND	–	Ground	V_{SS}, EV_{SS}, AV_{SS}	◎
CLK	Output	Clock output	–	×
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎
SI/RxD	Input	Receive signal	TOOL0	◎
SO/TxD	Output	Transmit signal		
SCK	Output	Transfer clock	–	×

Remark ◎: Be sure to connect the pin.
×: The pin does not have to be connected.

23.4 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.

<R>

23.4.1 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 **23.5 (1) Back ground event control register**). To pull it down externally, use a resistor of 200 kΩ or smaller.

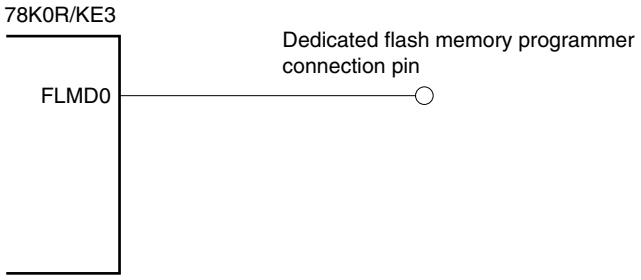
Self programming and the rewriting of flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V_{SS} pin.

(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 23-4. FLMD0 Pin Connection Example



23.4.2 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_{DD} via an external resistor.

When on-chip debugging is enabled in the normal operation mode, pull this pin up by connecting it to EV_{DD} 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).

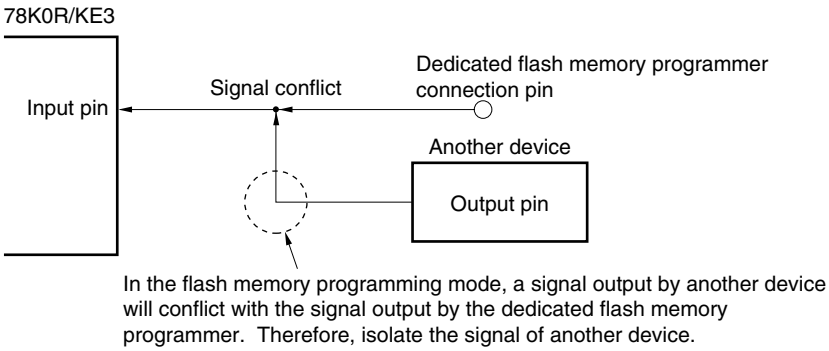
Remark The SAU and IIC0 pins are not used for communication between the 78K0R/KE3 and dedicated flash memory programmer, because single-line UART is used.

23.4.3 $\overline{\text{RESET}}$ 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 23-5. Signal Conflict ($\overline{\text{RESET}}$ Pin)



23.4.4 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.

23.4.5 REGC pin

Connect the REGC pin to GND via a capacitor (0.47 to 1 μF) in the same manner as during normal operation. 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.

23.4.6 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.

23.4.7 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_{DD}, EV_{SS}, AV_{REF}, and AV_{SS}) as those in the normal operation mode.

23.5 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 23-6. Format of Background Event Control Register (BECTL)

Address: FFFBEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BECTL	FLMDPUP	0	0	0	0	0	0	0

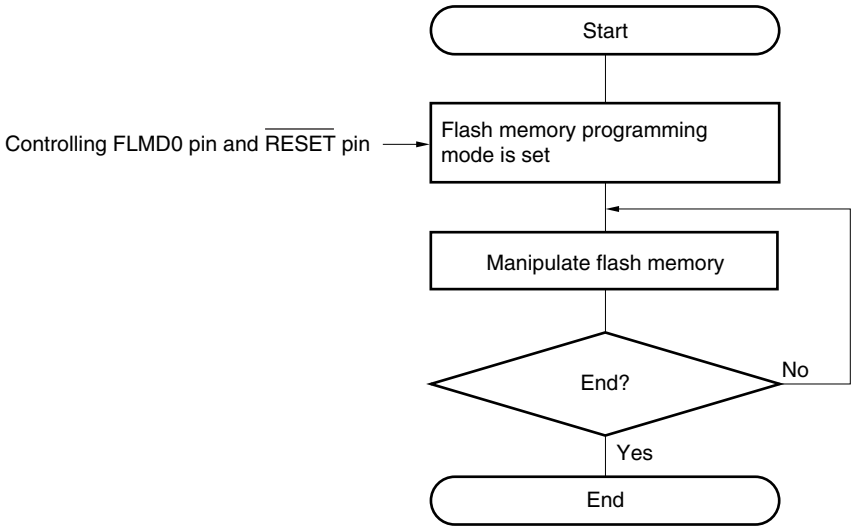
FLMDPUP	Software control of FLMD0 pin
0	Selects pull-down
1	Selects pull-up

23.6 Programming Method

23.6.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 23-7. Flash Memory Manipulation Procedure



23.6.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the 78K0R/KE3 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 23-8. Flash Memory Programming Mode

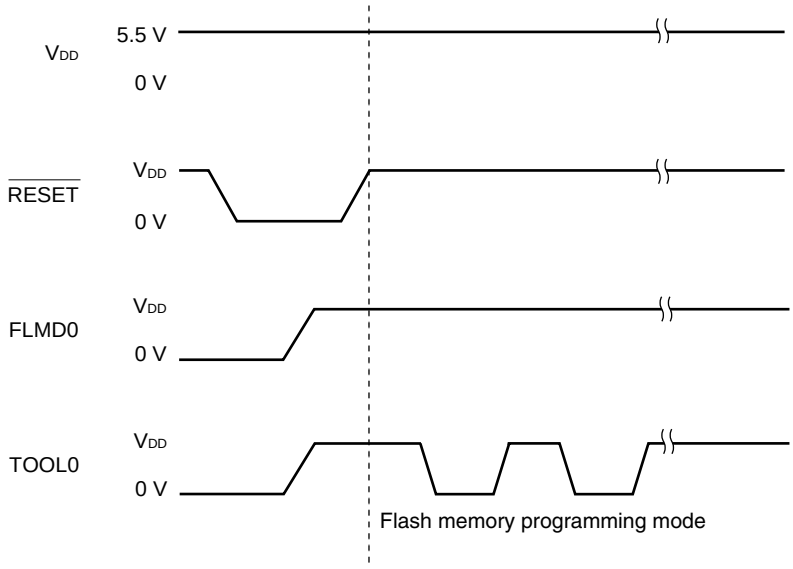


Table 23-3. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0	Normal operation mode
V_{DD}	Flash memory programming mode

23.6.3 Selecting communication mode

Communication mode of the 78K0R/KE3 as follows.

Table 23-4. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed	Frequency	Multiply Rate	
1-line mode (dedicated single-line UART)	UART-ch0	1 Mbps[Note 2]	–	–	TOOL0

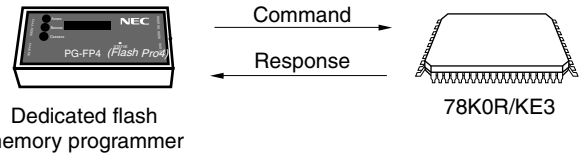
Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

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.

23.6.4 Communication commands

The 78K0R/KE3 communicates with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0R/KE3 are called commands, and the signals sent from the 78K0R/KE3 to the dedicated flash memory programmer are called response.

Figure 23-9. Communication Commands



The flash memory control commands of the 78K0R/KE3 are listed in the table below. All these commands are issued from the programmer and the 78K0R/KE3 perform processing corresponding to the respective commands.

Table 23-5. 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/KE3 information (such as the part number and flash memory configuration).
	Version Get	Gets the 78K0R/KE3 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/KE3 return a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0R/KE3 are listed below.

Table 23-6. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

23.7 Security Settings

The 78K0R/KE3 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 batch erase (chip erase) command, block erase command, and write command on boot cluster 0 in the flash memory is prohibited by this setting.

The batch erase (chip erase), block erase, write commands, and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by on-board/off-board programming and self programming. Each security setting can be used in combination.

All the security settings are cleared by executing the batch erase (chip erase) command.

Table 23-7 shows the relationship between the erase and write commands when the 78K0R/KE3 security function is enabled.

Remark To prohibit writing and erasing during self-programming, use the flash sealed window function (see **23.8.2** for detail).

Table 23-7. 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 sealed window function (see **23.8.2** for detail).

Table 23-8. 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		

23.8 Flash Memory Programming by Self-Programming

The 78K0R/KE3 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/KE3 self-programming library, it can be used to upgrade the program in the field.

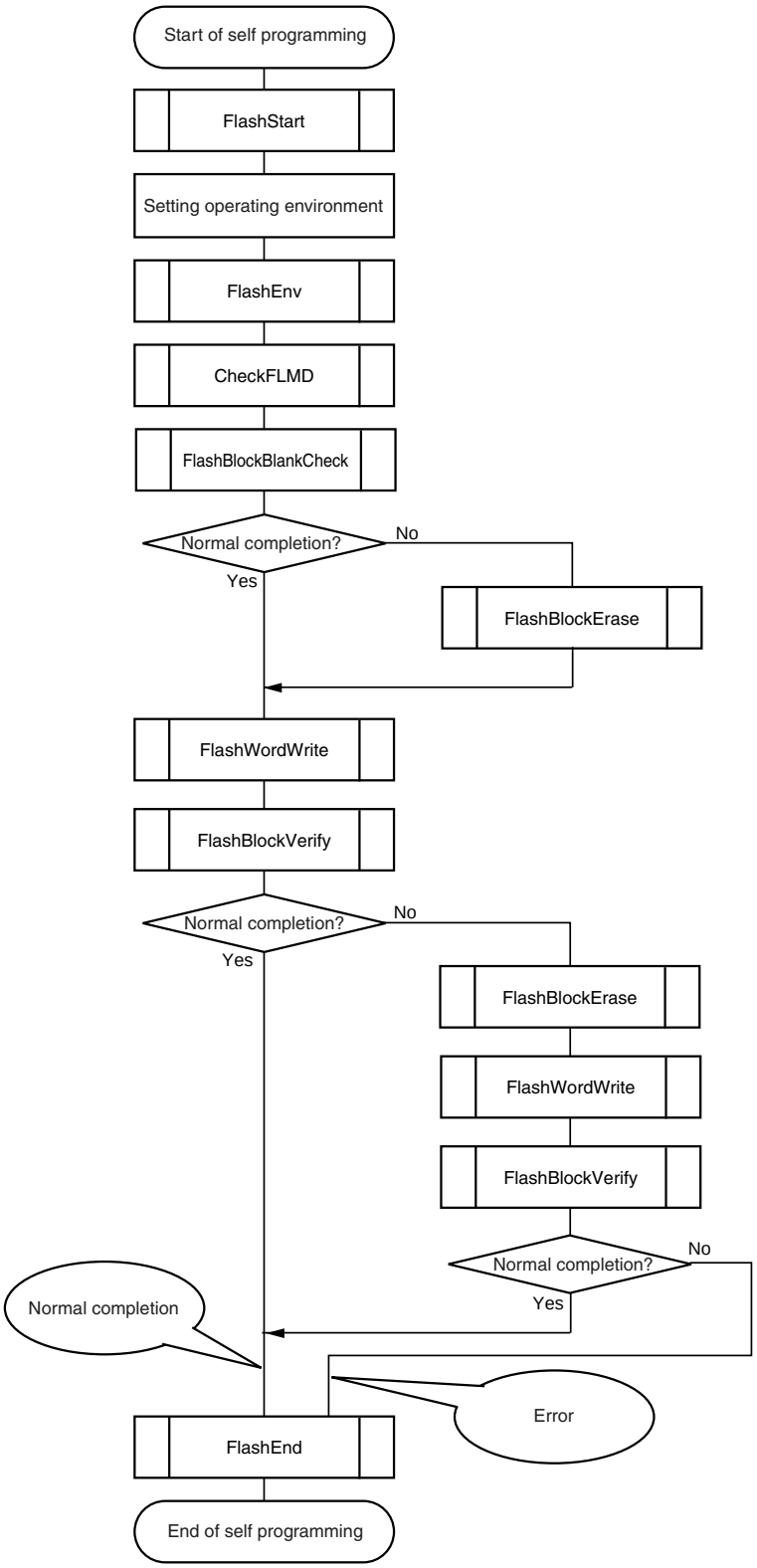
If an interrupt occurs during self-programming, self-programming can be temporarily stopped and interrupt servicing can be executed. If an unmasked interrupt request is generated in the EI state, the request branches directly from the self-programming library to the interrupt routine. After the self-programming mode is later restored, self-programming can be resumed. However, the interrupt response time is different from that of the normal operation mode.

Remark For details of the self-programming function and the 78K0R/KE3 self-programming library, refer to <R> **78K0R Microcontroller Self Programming Library Type01 User's Manual (U18706E)**.

Cautions
1. **The self-programming function cannot be used when the CPU operates with the subsystem clock.**
2. **In the self-programming mode, call the self-programming start library (FlashStart).**
3. **To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the self-programming library.**
4. **The self-programming function is disabled in the low consumption current mode. For details of the low consumption current mode, see CHAPTER 21 REGULATOR.**

The following figure illustrates a flow of rewriting the flash memory by using a self programming library.

<R> **Figure 23-10. Flow of Self Programming (Rewriting Flash Memory)**



Remark For details of the self programming library, refer to **78K0R Microcontroller Self Programming Library**
<R> **Type01 User's Manual (U18706E)**.

23.8.1 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.

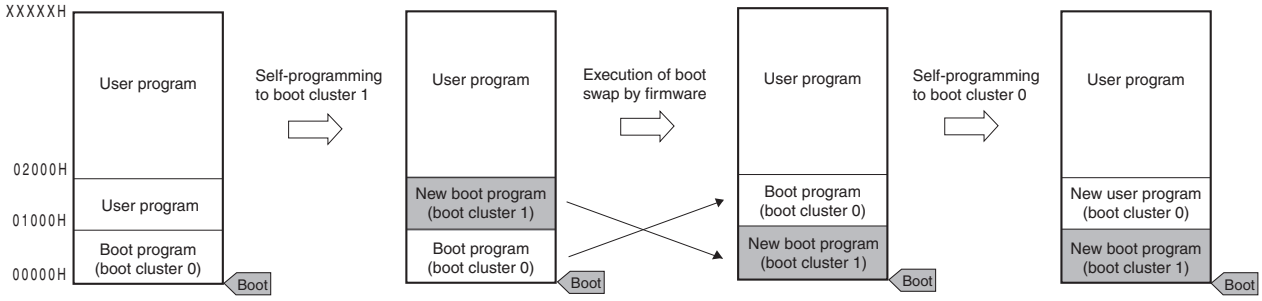
The boot swap function is used to avoid this problem.

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/KE3, 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 23-11. Boot Swap Function

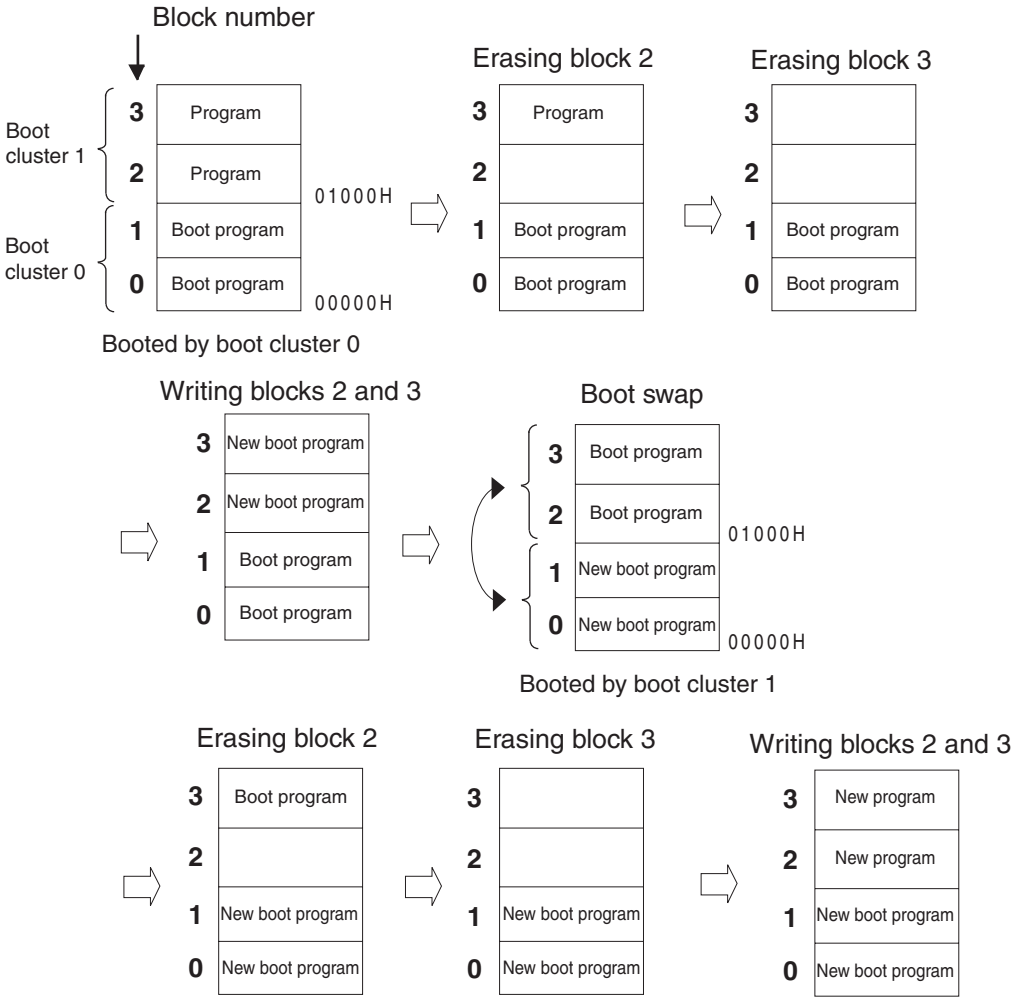


In an example of above figure, it is as follows.

Boot cluster 0: Boot program area before boot swap

Boot cluster 1: Boot program area after boot swap

Figure 23-12. Example of Executing Boot Swapping



23.8.2 Flash shield window function

The flash shield window function is provided as one of the security functions for self-programming.

Writing and erasing to the flash memory within the range specified as a window are enabled during self-programming, and writing and erasing to the flash memory outside the specified range are prohibited.

The window range can be expanded or reduced by setting and change during on-board/off-board programming and self-programming. However, the shield function becomes effective only during self-programming. In on-board/off-board programming, writing and erasing to the flash memory outside the window range are enabled.

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 23-9. 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 range of window range.
On-board/Off-board programming	Specify the starting and ending blocks on GUI of dedicated flash memory programmer, etc.	Block erasing is enabled also outside the window range.	Writing is enabled also outside the window range.

Remark See **23.7 Security Settings** to prohibit writing/erasing during on-board/off-board programming.

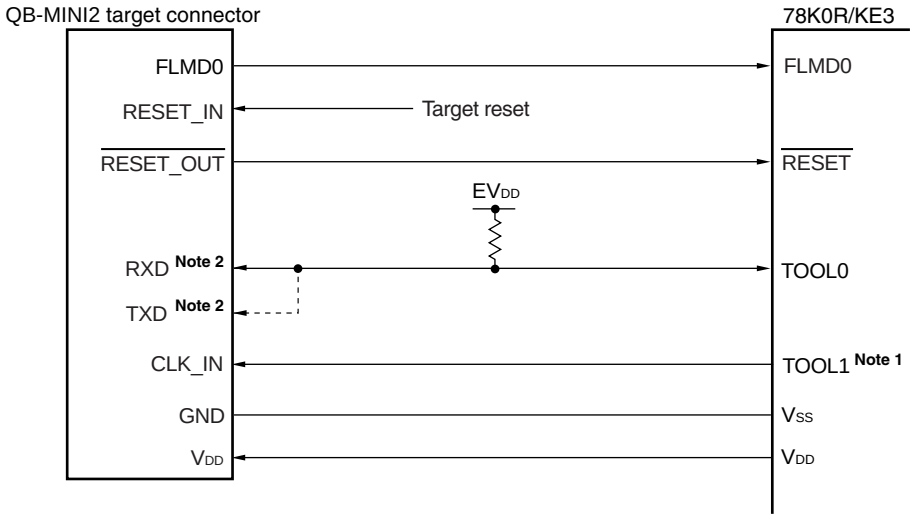
CHAPTER 24 ON-CHIP DEBUG FUNCTION

24.1 Connecting QB-MINI2 to 78K0R/KE3

The 78K0R/KE3 uses the V_{DD}, FLMD0, $\overline{RESET}$, TOOL0, TOOL1[Note], and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2).

Caution The 78K0R/KE3 has an on-chip debug function. Do not use this product for mass production because its reliability cannot be guaranteed after the on-chip debug function has been used, given the issue of the number of times the flash memory can be rewritten. NEC Electronics does not accept complaints concerning this product.

Figure 24-1. Connection Example of QB-MINI2 and 78K0R/KE3



Notes 1. Connection is not required for communication in 1-line mode but required for communication in 2-line mode. At this time, perform necessary connections according to Table 2-2 Connection of Unused Pins since TOOL1 is an unused pin when QB-MINI2 is unconnected.

2. Connecting the dotted line is not necessary since RXD and TXD are shorted within QB-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.

Remark The FLMD0 pin is recommended to be open for self-programming in on-chip debugging. To pull down externally, use a resistor of 100 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 24-1 lists the differences between 1-line mode and 2-line mode.

Table 24-1. Lists the Differences Between 1-line Mode and 2-line Mode.

Communication mode	Flash memory programming function	Debugging function
1-line mode	Available	• Pseudo real-time RAM monitor (RRM) function not supported. • DMM function (rewriting memory in RUN) not supported. • The debugger speed is two to four times slower than 2-line mode.
2-line mode	None	• Pseudo real-time RAM monitor (RRM) function supported • DMM function (rewriting memory in RUN) supported

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.

24.2 On-Chip Debug Security ID

The 78K0R/KE3 has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 22 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

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 24-2. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
010C4H to 010CDH	

24.3 Securing of user resources

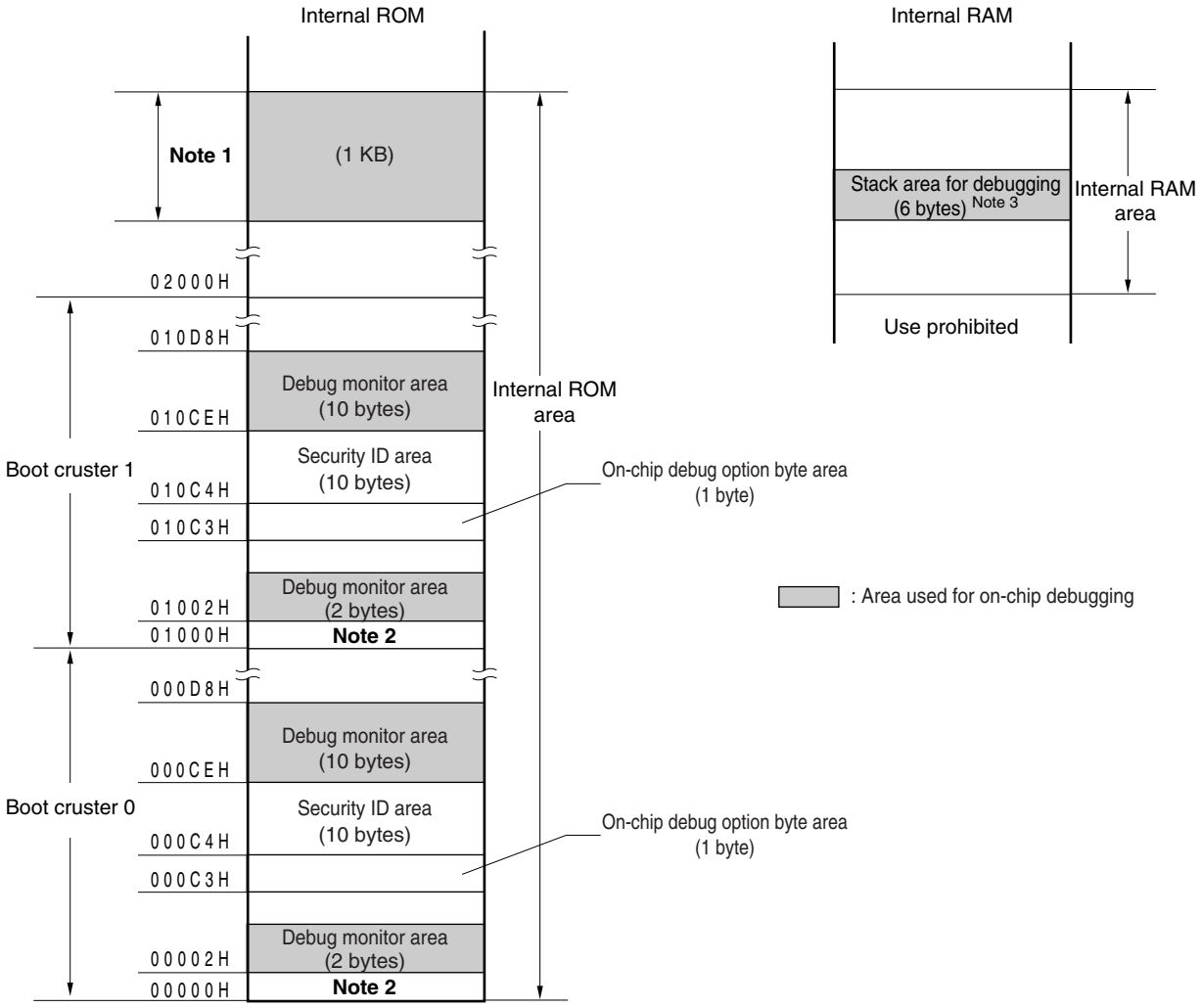
To perform communication between the 78K0R/KE3 and QB-MINI2, as well as each debug function, the securing of memory space must be done beforehand.

If NEC Electronics assembler RA78K0R or compiler CC78K0R is used, the items can be set by using linker options.

(1) Securement of memory space

The shaded portions in Figure 24-2 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 24-2. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on products as follows.

Products	Internal ROM	Address
μPD78F1142	64 KB	0FC00H-0FFFFH
μPD78F1143	96 KB	17C00H-17FFFH
μPD78F1144	128 KB	1FC00H-1FFFFH
μPD78F1145	192 KB	2FC00H-2FFFFH
μPD78F1146	256 KB	3FC00H-3FFFFH

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 25 BCD CORRECTION CIRCUIT

25.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.

The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/ subtracting the BCDADJ register.

25.2 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 25-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

25.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

<R> **(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 register.

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 Register	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 Register	AC Flag	BCDADJ Register
`MOV A, #85H ; <1>`	85H	–	–	–
`ADD A, #15H ; <2>`	9AH	0	0	06H
`ADD A, !BCDADJ ; <3>`	00H	1	1	–

Examples 3: 80 + 80 = 160

Instruction	A Register	CY Register	AC Flag	BCDADJ Register
`MOV A, #80H ; <1>`	80H	–	–	–
`ADD A, #80H ; <2>`	00H	1	0	60H
`ADD A, !BCDADJ ; <3>`	60H	1	0	–

<R> **(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 register.

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 Register	AC Flag	BCDADJ Register
`MOV A, #91H ; <1>`	91H	−	−	−
`SUB A, #52H ; <2>`	3FH	0	1	06H
`SUB A, !BCDADJ ; <3>`	39H	0	0	−

CHAPTER 26 INSTRUCTION SET

This chapter lists the instructions in the 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 26-5 Operation List** indicate the operation or instruction format that is newly added for the 78K0R microcontrollers.

26.1 Conventions Used in Operation List

26.1.1 Operand identifiers and specification methods

Operands are described in the "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Alphabetic letters in capitals and the symbols, #, !, !!, $, $!, [ ], and ES: are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: 16-bit absolute address specification
- !!: 20-bit absolute address specification
- $: 8-bit relative address specification
- $!: 16-bit relative address specification
- [ ]: Indirect address specification
- ES:: Extension address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, !!, $, $!, [ ], and ES: symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 26-1. Operand Identifiers and Specification Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol (SFR symbol)
sfrp	Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only[Note])
saddr	FFE20H to FFF1FH Immediate data or labels
saddrp	FFE20H to FF1FH Immediate data or labels (even addresses only[Note])
addr20	00000H to FFFFFH Immediate data or labels
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions[Note])
addr5	0080H to 00BFH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

Remark For special-function register symbol, see **Table 3-5 SFR List** and **Table 3-6 Extended SFR (2nd SFR) List**.

26.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 26-2. Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
∀	Exclusive logical sum (exclusive OR)
¯	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

26.1.3 Description of flag operation column

The change of the flag value when the instruction is executed is shown in the “Flag” column using the following symbols.

Table 26-3. Symbols in “Flag” Column

Symbol	Change of Flag Value
(Blank)	Unchanged
0	Cleared to 0
1	Set to 1
×	Set/cleared according to the result
R	Previously saved value is restored

26.1.4 PREFIX Instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

Table 26-4. Use Example of PREFIX Operation Code

Instruction	Opcode				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	–
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	–	–	–	–
MOV A, ES:[HL]	11H	8BH	–	–	–

Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.

26.2 Operation List

Table 26-5. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		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 or 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.

Table 26-5. 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 or 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.

Table 26-5. 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 or 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.

Table 26-5. 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 or 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.

Table 26-5. 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 or 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.

Table 26-5. 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 or 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.

Table 26-5. 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 or 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.

Table 26-5. 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 or 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.

Table 26-5. 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 or 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.

Table 26-5. 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 or 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.

Table 26-5. 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 or 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.

Table 26-5. 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 or 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.

3. cnt indicates the bit shift count.

Table 26-5. 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 or 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.

Table 26-5. 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 or 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.

Table 26-5. 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) ← (PC + 2)S, (SP – 3) ← (PC + 2)H, (SP – 4) ← (PC + 2)L, PC ← CS, rp, SP ← SP – 4			
		$!addr20	3	3	–	(SP – 2) ← (PC + 3)S, (SP – 3) ← (PC + 3)H, (SP – 4) ← (PC + 3)L, PC ← PC + 3 + jdisp16, SP ← SP – 4			
		!addr16	3	3	–	(SP – 2) ← (PC + 3)S, (SP – 3) ← (PC + 3)H, (SP – 4) ← (PC + 3)L, PC ← 0000, addr16, SP ← SP – 4			
		!!addr20	4	3	–	(SP – 2) ← (PC + 4)S, (SP – 3) ← (PC + 4)H, (SP – 4) ← (PC + 4)L, PC ← addr20, SP ← SP – 4			
	CALLT	[addr5]	2	5	–	(SP – 2) ← (PC + 2)S, (SP – 3) ← (PC + 2)H, (SP – 4) ← (PC + 2)L , PCS ← 0000, PCH ← (0000, addr5 + 1), PCL ← (0000, addr5), SP ← SP – 4			
	BRK	–	2	5	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 2)S, (SP – 3) ← (PC + 2)H, (SP – 4) ← (PC + 2)L, PCS ← 0000, PCH ← (0007FH), PCL ← (0007EH), SP ← SP – 4, IE ← 0			
	RET	–	1	6	–	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), SP ← SP + 4			
	RETI	–	2	6	–	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R
	RETB	–	2	6	–	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R

Notes 1. When the internal RAM area or 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.

Table 26-5. 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]	6/8	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/9	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1	

Notes 1. When the internal RAM area or 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.

Table 26-5. 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]	6/8	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/9	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	5/7[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 or 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.

3. n indicates the number of register banks (n = 0 to 3)

Caution The 78K0R/KE3 is provided with an on-chip debug function. After using the on-chip debug function, do not use the product for mass production because its reliability cannot be guaranteed from the viewpoint of the limit of the number of times the flash memory can be rewritten.
After the on-chip debug function is used, complaints will not be accepted.

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_{DD}		−0.5 to +6.5	V
		V_{SS}		−0.5 to +0.3	V
		EV_{SS}		−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
<R>	Input voltage	V_{I1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120 to P124, P140, P141, EXCLK, $\overline{\text{RESET}}$, FLMD0	−0.3 to EV_{DD} +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	−0.3 to AV_{REF} +0.3 and −0.3 to V_{DD} +0.3 Note 1	V
<R>	Output voltage	V_{O1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P130, P140, P141	−0.3 to EV_{DD} +0.3 Note 1	V
		V_{O2}	P20 to P27	−0.3 to AV_{REF} +0.3	V
	Analog input voltage	V_{AN}	ANI0 to ANI7	−0.3 to AV_{REF} +0.3 Note 1 and −0.3 to V_{DD} +0.3 Note 1	V

Notes 1. Must be 6.5 V or lower.
2. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

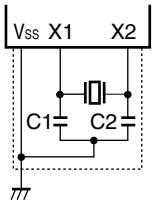
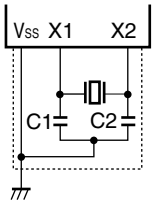
Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P130, P140, P141	−10	mA
		Total of all pins −80 mA	P00 to P04, P40 to P43, P120, P130, P140, P141	−25	mA
			P05, P06, P10 to P17, P30, P31, P50 to P55, P70 to P77	−55	mA
	I_{OH2}	Per pin	P20 to P27	−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 P43, P50 to P55, P60 to P63, P70 to P77, P120, P130, P140, P141	30	mA
		Total of all pins 200 mA	P00 to P04, P40 to P43, P120, P130, P140, P141	60	mA
			P05, P06, P10 to P17, P30, P31, P50 to P55, P60 to P63, P70 to P77	140	mA
	I_{OL2}	Per pin	P20 to P27	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

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

X1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		X1 clock oscillation frequency (f_X)[Note]	2.7 V ≤ V_{DD} ≤ 5.5 V	2.0		20.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V	2.0		5.0	
Crystal resonator		X1 clock oscillation frequency (f_X)[Note]	2.7 V ≤ V_{DD} ≤ 5.5 V	2.0		20.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V	2.0		5.0	

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. Since the CPU is started by the internal high-speed oscillation clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Internal Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
8 MHz internal oscillator	Internal high-speed oscillation clock frequency (f_{IH})[Note 1]	2.7 V ≤ V_{DD} ≤ 5.5 V		7.6	8.0	8.4	MHz
		1.8 V ≤ V_{DD} < 2.7 V		5.0	8.0	8.4	MHz
240 kHz internal oscillator	Internal low-speed oscillation clock frequency (f_{IL})	Normal current mode	2.7 V ≤ V_{DD} ≤ 5.5 V	216	240	264	kHz
			1.8 V ≤ V_{DD} < 2.7 V	192	240	264	kHz
		Low consumption current mode[Note 2]		192	240	264	kHz

<R> **Notes 1.** This only indicates the oscillator characteristics of when HIOTRM is set to 10H. Refer to AC Characteristics for instruction execution time.

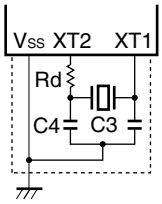
2. Regulator output is set to low consumption current mode in the following cases:

- When the RMC register is set to 5AH.
- During system reset
- In STOP mode (except during OCD mode)
- When both the high-speed system clock (f_{MX}) and the high-speed internal oscillation clock (f_{IH}) are stopped during CPU operation with the subsystem clock (f_{XT})
- When both the high-speed system clock (f_{MX}) and the high-speed internal oscillation clock (f_{IH}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{XT}) has been set.

Remark For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 21 REGULATOR**.

XT1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 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.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

DC Characteristics (1/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = 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 P43, P50 to P55, P70 to P77, P120, P130, P140, P141	4.0 V ≤ V_{DD} ≤ 5.5 V			−3.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			−1.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			−1.0	mA
		Total of P00 to P04, P40 to P43, P120, P130, P140, P141 (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
			1.8 V ≤ V_{DD} < 2.7 V			−5.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P55, P70 to P77 (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
			1.8 V ≤ V_{DD} < 2.7 V			−10.0	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
			1.8 V ≤ V_{DD} < 2.7 V			−15.0	mA
<R>	I_{OH2}	Per pin for P20 to P27	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_{DD} 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 I_{OH} = 20.0 mA and n = 50%

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 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (2/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = 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 P43, P50 to P55, P70 to P77, P120, P130, P140, P141	4.0 V ≤ V_{DD} ≤ 5.5 V			8.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			1.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			0.5	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
			1.8 V ≤ V_{DD} < 2.7 V			2.0	mA
		Total of P00 to P04, P40 to P43, P120, P130, P140, P141 (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
			1.8 V ≤ V_{DD} < 2.7 V			9.0	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P55, P60 to P63, P70 to P77 (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
			1.8 V ≤ V_{DD} < 2.7 V			20.0	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
			1.8 V ≤ V_{DD} < 2.7 V			29.0	mA
<R>	I_{OL2}	Per pin for P20 to P27	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_{SS}, 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 I_{OL} = 20.0 mA and n = 50%

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 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (3/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P01, P02, P12, P13, P15, P41, P52 to P55, P121 to P124		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P00, P03 to P06, P10, P11, P14, P16, P17, P30, P31, P40, P42, P43, P50, P51, P70 to P77, P120, P140, P141, EXCLK, $\overline{RESET}$	Normal input buffer	0.8V_{DD}		V_{DD}	V
	V_{IH3}	P03, P04	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
			TTL input buffer 1.8 V ≤ V_{DD} < 2.7 V	1.6		V_{DD}	V
<R>	V_{IH4}	P20 to P27	2.7 V ≤ AV_{REF} ≤ V_{DD}	0.7AV_{REF}		AV_{REF}	V
			AV_{REF} = V_{DD} < 2.7 V				
	V_{IH5}	P60 to P63		0.7V_{DD}		6.0	V
	V_{IH6}	FLMD0		0.9V_{DD} Note 1		V_{DD}	V
Input voltage, low	V_{IL1}	P01, P02, P12, P13, P15, P41, P52 to P55, P121 to P124		0		0.3V_{DD}	V
	V_{IL2}	P00, P03 to P06, P10, P11, P14, P16, P17, P30, P31, P40, P42, P43, P50, P51, P70 to P77, P120, P140, P141, EXCLK, $\overline{RESET}$	Normal input buffer	0		0.2V_{DD}	V
	V_{IL3}	P03, P04	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
			TTL input buffer 1.8 V ≤ V_{DD} < 2.7 V	0		0.2	V
<R>	V_{IL4}	P20 to P27	2.7 V ≤ AV_{REF} ≤ V_{DD}	0		0.3AV_{REF}	V
			AV_{REF} = V_{DD} < 2.7 V				
	V_{IL5}	P60 to P63		0		0.3V_{DD}	V
	V_{IL6}	FLMD0 Note 2		0		0.1V_{DD}	V

Notes 1. The high-level input voltage (V_{IH6}) must be greater than 0.9V_{DD} when using it in the flash memory programming mode.

2. 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}.

Cautions 1. The maximum value of V_{IH} of pins P02 to P04 is V_{DD}, even in the N-ch open-drain mode.

<R> **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.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (4/10)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = 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 P43, P50 to P55, P70 to P77, P120, P130, P140, P141	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = – 3.0 mA	V_{DD} – 0.7			V
				1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –1.0 mA	V_{DD} – 0.5			V
<R>		V_{OH2}	P20 to P27	AV_{REF} ≤ V_{DD}, I_{OH2} = –0.1 mA	AV_{REF} – 0.5			V
	Output voltage, low	V_{OL1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120, P130, P140, P141	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
				1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.5 mA			0.4	V
<R>		V_{OL2}	P20 to P27	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
				1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 2.0 mA			0.4	V

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (5/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = 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 P43, P50 to P55, P60 to P63, P70 to P77, P120, P140, P141, FLMD0, $\overline{RESET}$	$V_I = V_{DD}$				1	μA
<R>	I_{LIH2}	P20 to P27	$V_I = AV_{REF}$, 2.7 V ≤ AV_{REF} ≤ V_{DD}				1	μA
			$V_I = AV_{REF}$, AV_{REF} = V_{DD} < 2.7 V					
	I_{LIH3}	P121 to P124 (X1, X2, XT1, XT2)	$V_I = V_{DD}$	In input port			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P140, P141, FLMD0, $\overline{RESET}$	$V_I = V_{SS}$				−1	μA
<R>	I_{LIL2}	P20 to P27	$V_I = V_{SS}$, 2.7 V ≤ AV_{REF} ≤ V_{DD}				−1	μA
			$V_I = V_{SS}$, AV_{REF} = V_{DD} < 2.7 V					
	I_{LIL3}	P121 to P124 (X1, X2, XT1, XT2)	$V_I = V_{SS}$	In input port			−1	μA
				In resonator connection			−10	μA

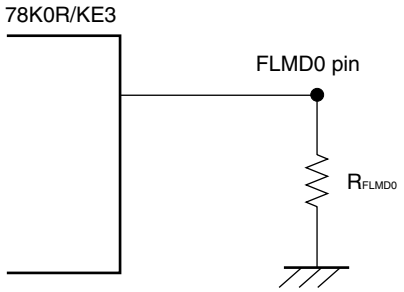
Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (6/10)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = 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 P43, P50 to P55, P70 to P77, P120, P140, P141	V_I = V_{SS}, in input port	10	20	100	kΩ
FLMD0 pin external pull-down resistance [Note]	R_{FLMD0}	When enabling the self-programming mode setting with software		100			kΩ

Note It is recommended to leave the FLMD0 pin open. If the pin is required to be pulled down externally, set R_{FLDM0} to 100 kΩ or more.



Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (7/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
<R> Supply current	I_{DD1} Note 1	Operating mode	f_{MX} = 20 MHz Note 2, V_{DD} = 5.0 V	Square wave input		8.2	12.2	mA
				Resonator connection		8.5	12.5	mA
			f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V	Square wave input		8.2	12.2	mA
				Resonator connection		8.5	12.5	mA
			f_{MX} = 10 MHz Notes 2, 3, V_{DD} = 5.0 V	Square wave input		3.9	6.2	mA
				Resonator connection		4.0	6.3	mA
			f_{MX} = 10 MHz Notes 2, 3, V_{DD} = 3.0 V	Square wave input		3.9	6.2	mA
				Resonator connection		4.0	6.3	mA
			f_{MX} = 5 MHz Notes 2, 3, V_{DD} = 3.0 V	Normal current mode, Square wave input		2.1	3.0	mA
				Normal current mode, Resonator connection		2.2	3.1	mA
<R>				Low consumption current mode Note 4, Square wave input		1.5	2.1	mA
				Low consumption current mode Note 4, Resonator connection		1.5	2.1	mA
			f_{MX} = 5 MHz Notes 2, 3, V_{DD} = 2.0 V	Normal current mode, Square wave input		1.5	2.1	mA
				Normal current mode, Resonator connection		1.5	2.1	mA
<R>				Low consumption current mode Note 4, Square wave input		1.4	2.0	mA
				Low consumption current mode Note 4, Resonator connection		1.4	2.0	mA
			f_{IH} = 8 MHz Note 5	V_{DD} = 5.0 V		3.5	5.0	mA
				V_{DD} = 3.0 V		3.5	5.0	mA
			f_{SUB} = 32.768 kHz Note 6, T_A = −40 to +70 °C	V_{DD} = 5.0 V		8.0	24.0	μA
				V_{DD} = 3.0 V		8.0	24.0	μA
				V_{DD} = 2.0 V		7.0	21.0	μA
			f_{SUB} = 32.768 kHz Note 6, T_A = −40 to +85 °C	V_{DD} = 5.0 V		8.0	31.0	μA
				V_{DD} = 3.0 V		8.0	31.0	μA
				V_{DD} = 2.0 V		7.0	28.0	μA

Notes 1. Total current flowing into V_{DD}, EV_{DD}, 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 and subsystem clock are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0 and FSEL (bit 0 of operation speed mode control register (OSMC)) = 0.

<R> **4.** When the RMC register is set to 5AH.

5. When high-speed system clock and subsystem clock are stopped.

6. When internal high-speed oscillator and high-speed system clock are stopped. When watchdog timer is stopped.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

<R> **4.** For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 21 REGULATOR**.

DC Characteristics (8/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

	Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
<R>	Supply current	I_{DD2} Note 1	HALT mode	f_{MX} = 20 MHz Note 2, V_{DD} = 5.0 V	Square wave input		1.1	2.7	mA
					Resonator connection		1.4	3.0	mA
				f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V	Square wave input		1.1	2.7	mA
					Resonator connection		1.4	3.0	mA
				f_{MX} = 10 MHz Notes 2, 3, V_{DD} = 5.0 V	Square wave input		0.65	1.4	mA
					Resonator connection		0.75	1.5	mA
				f_{MX} = 10 MHz Notes 2, 3, V_{DD} = 3.0 V	Square wave input		0.65	1.4	mA
					Resonator connection		0.75	1.5	mA
				f_{MX} = 5 MHz Notes 2, 3, V_{DD} = 3.0 V, Normal current mode	Square wave input		0.39	0.75	mA
					Resonator connection		0.44	0.8	mA
<R>				Low consumption current mode Note 4	Square wave input		0.3	0.5	mA
					Resonator connection		0.35	0.55	mA
				f_{MX} = 5 MHz Notes 2, 3, V_{DD} = 2.0 V, Normal current mode	Square wave input		0.3	0.5	mA
					Resonator connection		0.35	0.55	mA
<R>				Low consumption current mode Note 4	Square wave input		0.3	0.5	mA
					Resonator connection		0.35	0.55	mA
				f_{IH} = 8 MHz Note 5	V_{DD} = 5.0 V		0.45	0.6	mA
					V_{DD} = 3.0 V		0.45	0.6	mA

Notes 1. Total current flowing into V_{DD}, EV_{DD}, 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 and subsystem clock are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0 and FSEL (bit 0 of operation speed mode control register (OSMC)) = 0.

<R> **4.** When the RMC register is set to 5AH.

5. When high-speed system clock and subsystem clock are stopped.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: Internal high-speed oscillation clock frequency

<R> **3.** For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 21 REGULATOR**.

DC Characteristics (9/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = 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 +70 °C	V_{DD} = 5.0 V		2.2	14.0	μA
				V_{DD} = 3.0 V		2.2	14.0	μA
				V_{DD} = 2.0 V		2.1	13.8	μA
			f_{SUB} = 32.768 kHz Note 2, T_A = −40 to +85 °C	V_{DD} = 5.0 V		2.2	21.0	μA
				V_{DD} = 3.0 V		2.2	21.0	μA
				V_{DD} = 2.0 V		2.1	20.8	μA
	I_{DD3} Note 3	STOP mode	T_A = −40 to +70 °C			1.1	9.0	μA
			T_A = −40 to +85 °C			1.1	16.0	μA

Notes 1. Total current flowing into V_{DD}, EV_{DD}, 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 and high-speed system clock are stopped. When watchdog timer is stopped.

3. Total current flowing into V_{DD}, EV_{DD}, and AV_{REF}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. When subsystem clock is stopped. When watchdog timer is stopped.

Remark f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

DC Characteristics (10/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = 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
			V_{DD} = 2.0 V		0.2	1.0	
Watchdog timer operating current	I_{WDT} Notes 2, 3	f_{IL} = 240 kHz			5	10	μA
A/D converter operating current	I_{ADC} Note 4	During conversion at maximum speed, 2.3 V ≤ AV_{REF}			0.86	1.9	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 current value of the 78K0R/KE3 is the TYP. value, the sum of the TYP. values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time counter operates in operation mode or HALT mode. The I_{DD1} and I_{DD2} MAX. values also include the real-time counter operating current.

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 240 kHz internal oscillator). The current value of the 78K0R/KE3 is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when f_{CLK} = f_{SUB}/2 or when the watchdog timer operates in STOP mode.

4. Current flowing only to the A/D converter (AV_{REF} pin). The current value of the 78K0R/KE3 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/KE3 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. f_{IL}: Internal low-speed oscillation clock frequency

2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

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

AC Characteristics

(1) Basic operation (1/6)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

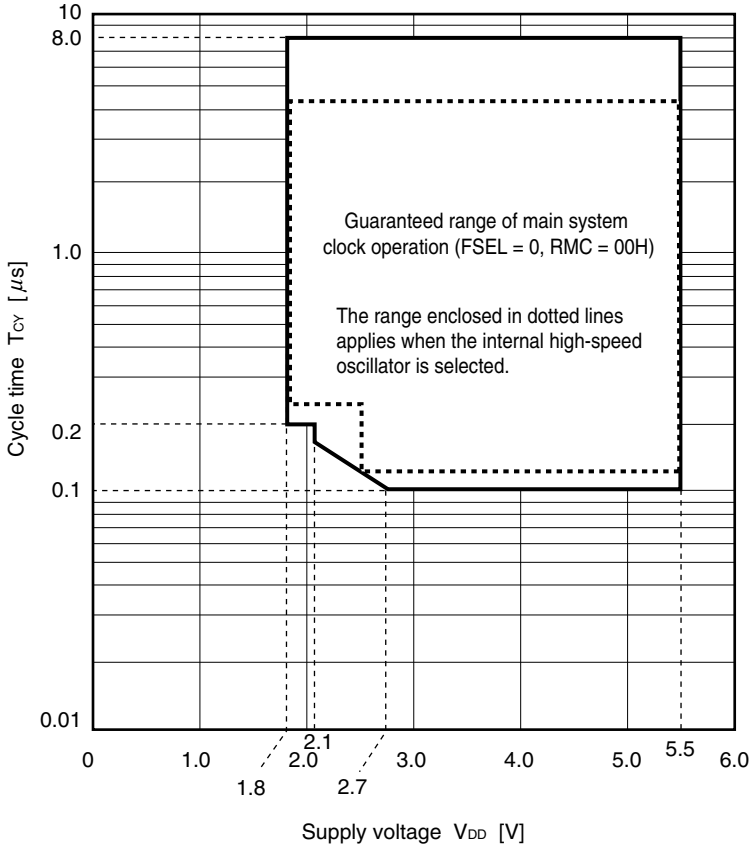
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{XP}) operation	Normal current mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.05		8	μs
				1.8 V ≤ V_{DD} < 2.7 V	0.2		8	μs
			Low consumption current mode		0.2		8	μs
		Subsystem clock (f_{SUB}) operation			57.2	61	62.5	μs
		In the self programming mode	Normal current mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.05		0.5	μs
External main system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V			2.0		20.0	MHz
		1.8 V ≤ V_{DD} < 2.7 V			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			24			ns
		1.8 V ≤ V_{DD} < 2.7 V			96			ns
TI00 to TI06 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK} + 10			ns
TO00 to TO06 output frequency	f_{TO}	2.7 V ≤ V_{DD} ≤ 5.5 V					10	MHz
		1.8 V ≤ V_{DD} < 2.7 V					5	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	2.7 V ≤ V_{DD} ≤ 5.5 V					10	MHz
		1.8 V ≤ V_{DD} < 2.7 V					5	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}: Timer array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of the TMR0n register. n: Channel number (n = 0 to 6))

2. For details on the normal current mode and low consumption current mode according to the regulator output voltage, refer to **CHAPTER 21 REGULATOR**.

(1) Basic operation (2/6)

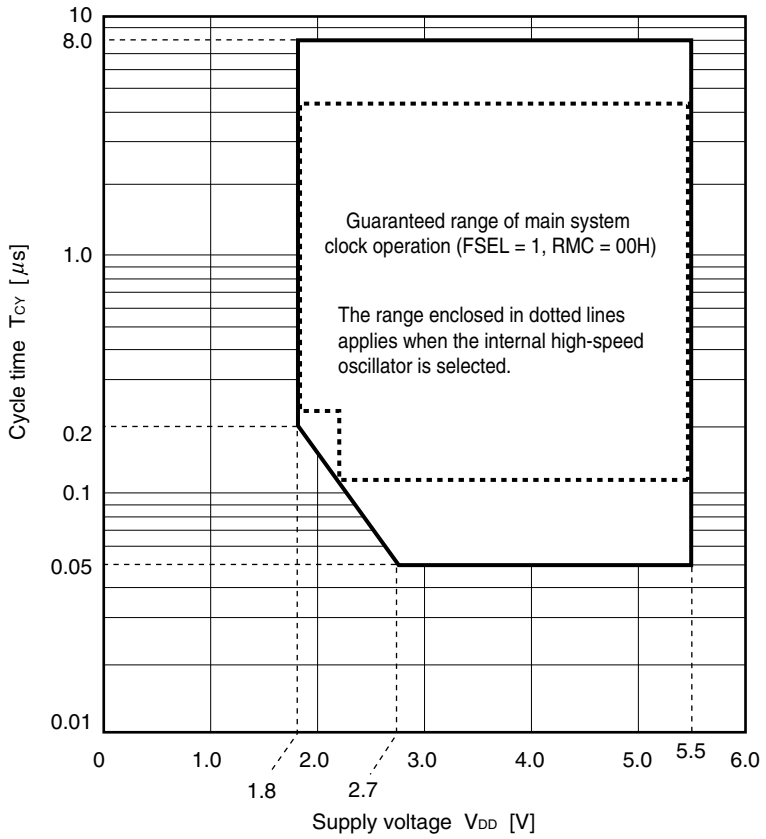
<R> **Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 00H)**



Remark FSEL: Bit 0 of the operation speed mode control register (OSMC)

(1) Basic operation (3/6)

<R> **Minimum instruction execution time during main system clock operation (FSEL = 1, RMC = 00H)**



Caution The following operations are prohibited when V_{DD} is less than 2.25 V.

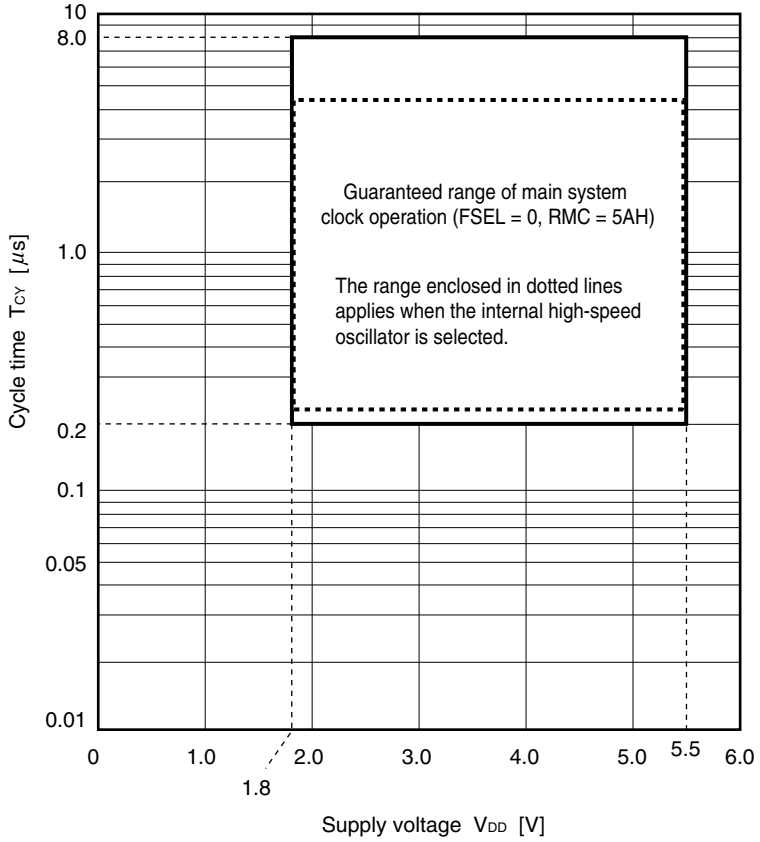
- **Operation rewriting FSEL from 0 to 1**
- **Releasing STOP mode during f_{EX} operation and f_{IH} operation, when FSEL is set to 1**
  (This must not be performed even if the frequency is divided. The STOP mode may be released during f_X operation.)
- **Operation to switch f_{CLK} from f_{SUB} to f_{MAIN}, while FSEL = 1**
  (This must not be performed even if the frequency is divided.)

Remarks 1. FSEL: Bit 0 of the operation speed mode control register (OSMC)

2. f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency
f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency

(1) Basic operation (4/6)

<R> **Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 5AH)**

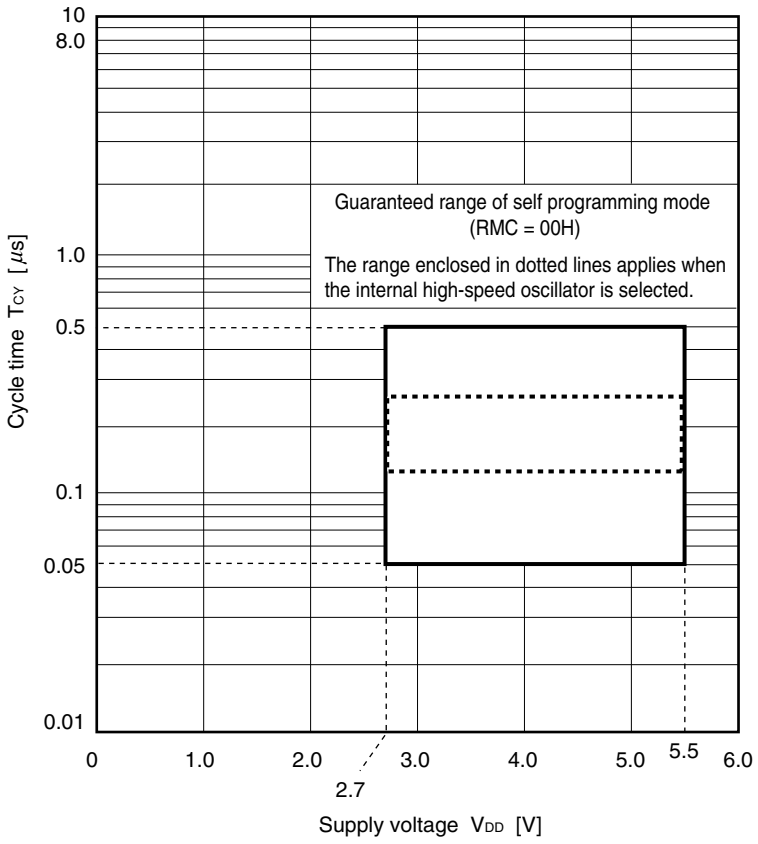


Remarks 1. FSEL: Bit 0 of the operation speed mode control register (OSMC)

2. The entire voltage range is 5 MHz (MAX.) when RMC is set to 5AH.

(1) Basic operation (5/6)

<R> **Minimum instruction execution time during self programming mode (RMC = 00H)**

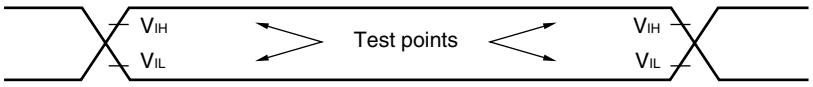


Remarks 1. FSEL: Bit 0 of the operation speed mode control register (OSMC)

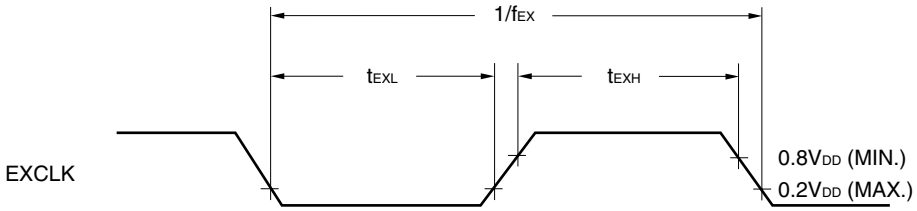
2. The self programming function cannot be used when RMC is set to 5AH or the CPU operates with the subsystem clock.

(1) Basic operation (6/6)

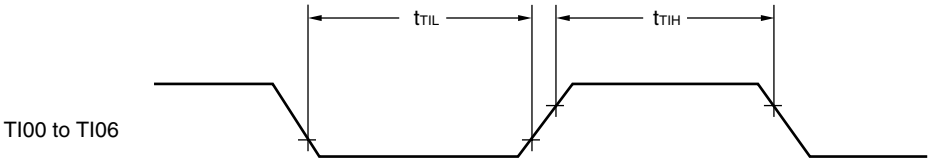
<R> **AC Timing Test Points**



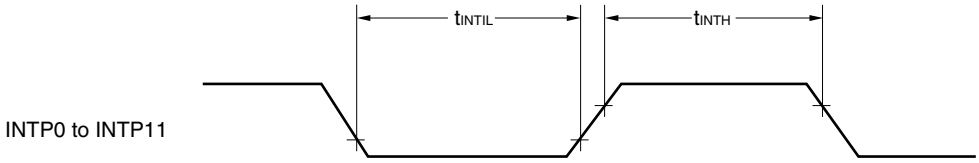
External Main System Clock Timing



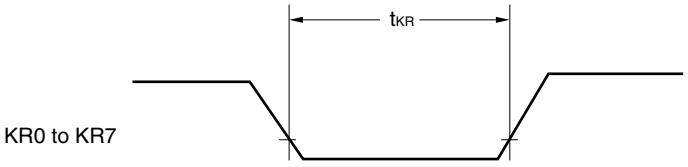
TI Timing



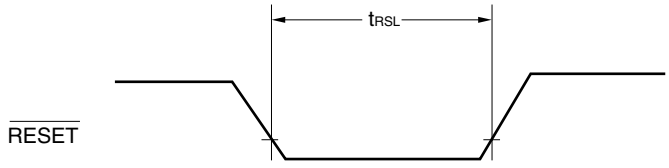
Interrupt Request Input Timing



Key Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing



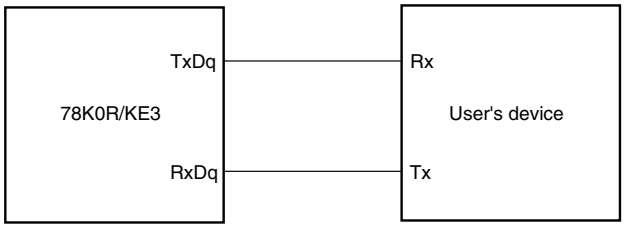
(2) Serial interface: Serial array unit (1/17)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

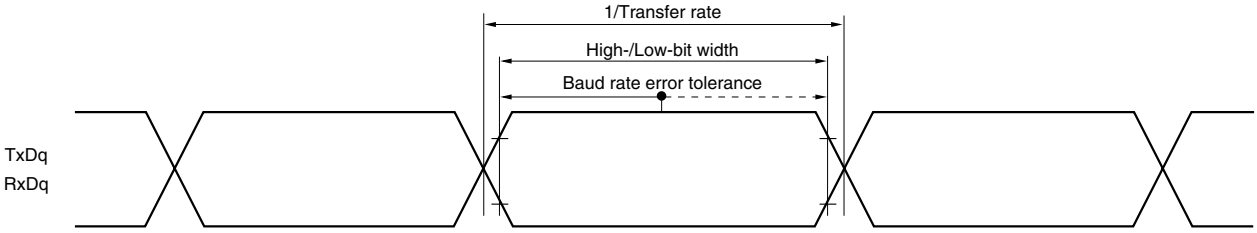
(a) During communication at same potential (UART mode) (dedicated baud rate generator output)

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 When using UART1, select the normal input buffer for RxD1 and the normal output mode for TxD1 by using the PIM0 and POM0 registers.

Remarks 1. q: UART number (q = 0, 1, 3)

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))

(2) Serial interface: Serial array unit (2/17)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 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			ns
		2.7 V ≤ V_{DD} < 4.0 V	400			ns
		1.8 V ≤ V_{DD} < 2.7 V	800			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
		1.8 V ≤ V_{DD} < 2.7 V	t_{KCY1}/2 – 80			ns
SIp setup time (to $\overline{SCKp}$↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V	70			ns
		2.7 V ≤ V_{DD} < 4.0 V	100			ns
		1.8 V ≤ V_{DD} < 2.7 V	190			ns
SIp hold time (from $\overline{SCKp}$↑) Note 2	t_{KSI1}		30			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 3	t_{KSO1}	C = 50 pF Note 4			40	ns

Notes 1. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

3. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

4. C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

Caution When using CSI10, select the normal input buffer for SI10 and the normal output mode for SO10 and $\overline{SCK10}$ by using the PIM0 and POM0 registers.

Remark p: CSI number (p = 00, 10), n: Channel number (n = 0, 2)

(2) Serial interface: Serial array unit (3/17)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 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}	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}			1/f_{MCK}+80			ns
SIp hold time (from $\overline{SCKp}$↑) Note 2	t_{KSI2}			50			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 3	t_{KSO2}	C = 50 pF Note 4	4.0 V ≤ V_{DD} ≤ 5.5 V			1/f_{MCK} + 120	ns
			2.7 V ≤ V_{DD} < 4.0 V			1/f_{MCK} + 120	ns
			1.8 V ≤ V_{DD} < 2.7 V			1/f_{MCK} + 180	ns

Notes 1. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

3. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

4. C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

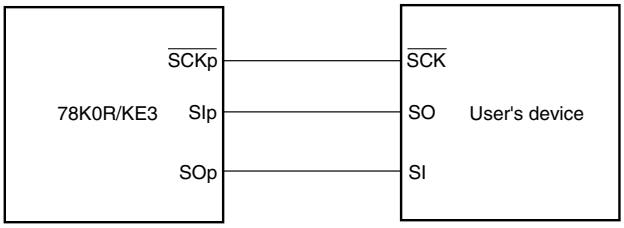
Caution When using CSI10, select the normal input buffer for SI10 and $\overline{SCK10}$ and the normal output mode for SO10 by using the PIM0 and POM0 registers.

Remarks 1. p: CSI number (p = 00, 10)

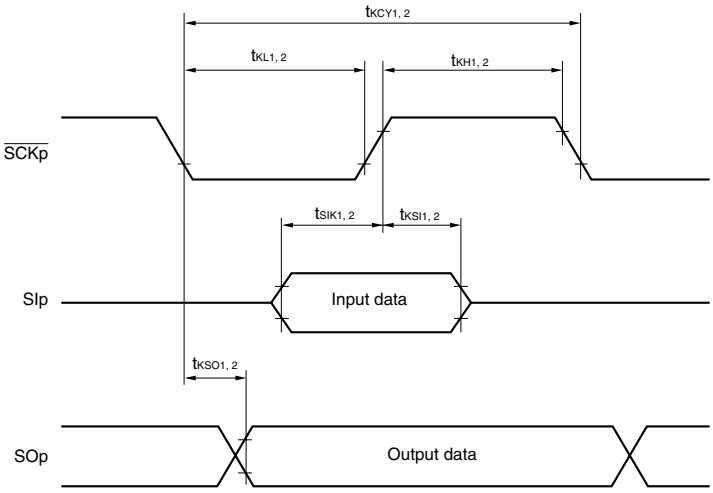
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of the SMR0n register. n: Channel number (n = 0, 2))

(2) Serial interface: Serial array unit (4/17)

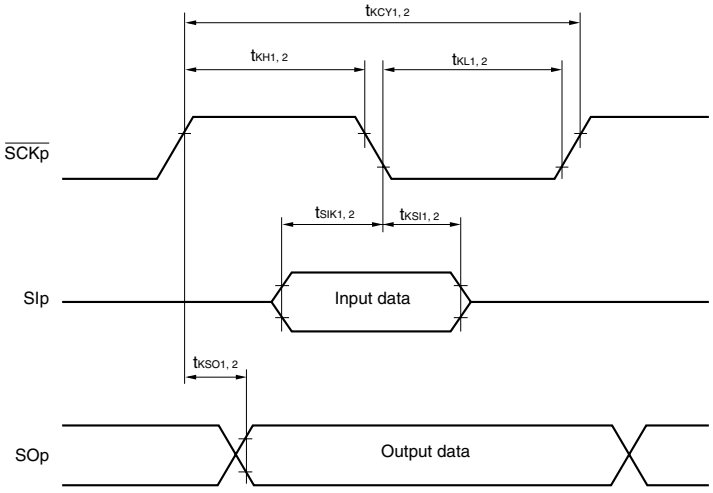
CSI mode connection diagram (during communication at same potential)



CSI mode serial transfer timing (during communication at same potential)
(When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1.)



CSI mode serial transfer timing (during communication at same potential)
(When DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.)



Remarks 1. p: CSI number (p = 00, 10)

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

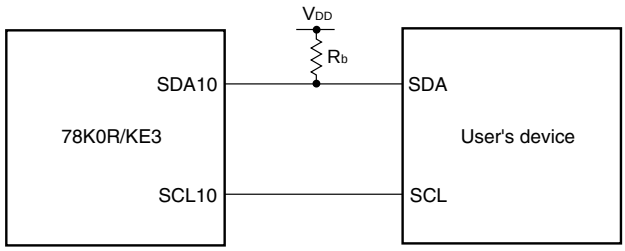
(2) Serial interface: Serial array unit (5/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

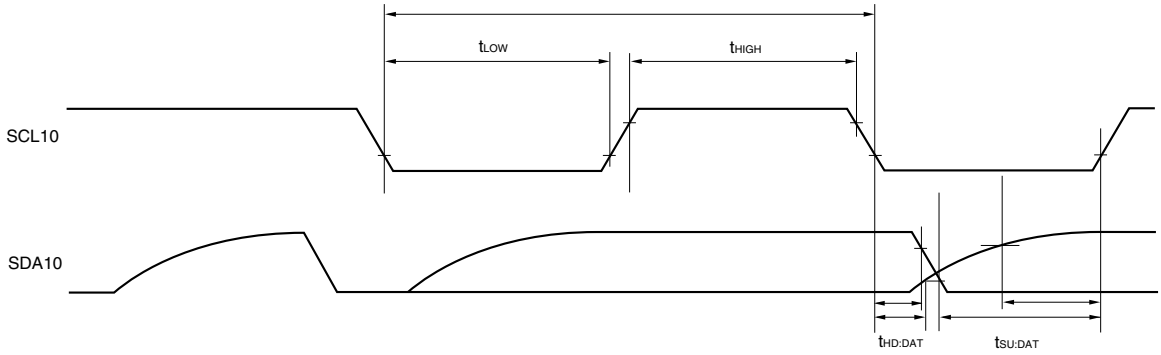
(d) During communication at same potentia (simplified I^2C mode)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCL10 clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		400	kHz
Hold time when SCL10 = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	995		ns
Hold time when SCL10 = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	995		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	160	ns

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 SDA10 and the normal output mode for SCL10 by using the PIM0 and POM0 registers.

Remarks 1. R_b[Ω]:Communication line (SDA10) pull-up resistance,
C_b[F]: Communication line (SCL10, SDA10) load capacitance

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS02 bit of the SMR02 register.)

(2) Serial interface: Serial array unit (6/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 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 RxD1 and the N-ch open drain output (V_{DD} tolerance) mode for TxD1 by using the PIM0 and POM0 registers.

Remarks 1. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of the SMR0n register. n: Channel number (n = 2, 3))

2. 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

3. UART0 and UART3 cannot communicate at different potential. Use UART1 for communication at different potential.

(2) Serial interface: Serial array unit (7/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 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 RxD1 and the N-ch open drain output (V_{DD} tolerance) mode for TxD1 by using the PIM0 and POM0 registers.

(Remark are given on the next page.)

(2) Serial interface: Serial array unit (8/17)

Remarks 1. R_b[Ω]:Communication line (TxD1) pull-up resistance,
C_b[F]: Communication line (TxD1) load capacitance, V_b[V]: Communication line voltage

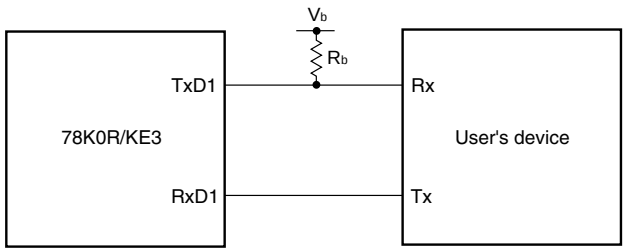
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of the SMR0n register. n: Channel number (n = 2, 3))

3. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.
$4.0\ V \le V_{DD} \le 5.5\ V$, $2.7\ V \le V_b \le 4.0\ V$: $V_{IH} = 2.2\ V$, $V_{IL} = 0.8\ V$
$2.7\ V \le V_{DD} \le 4.0\ V$, $2.3\ V \le V_b \le 2.7\ V$: $V_{IH} = 2.0\ V$, $V_{IL} = 0.5\ V$

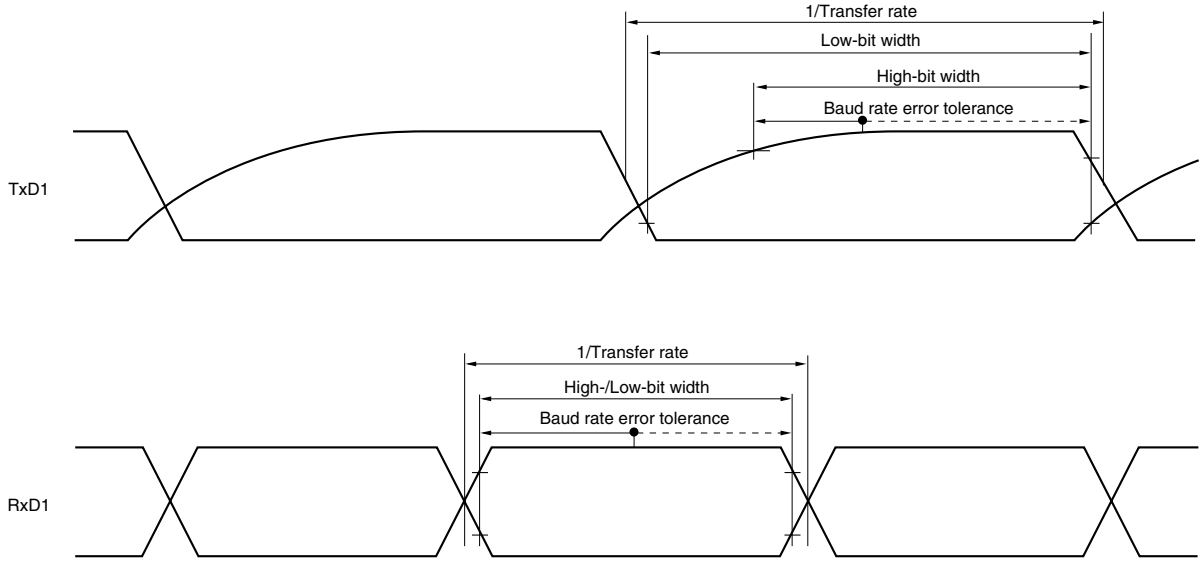
4. UART0 and UART3 cannot communicate at different potential. Use UART1 for communication at different potential.

(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 RxD1 and the N-ch open drain output (V_{DD} tolerance) mode for TxD1 by using the PIM0 and POM0 registers.

Remarks 1. R_b[Ω]:Communication line (TxD1) pull-up resistance, V_b[V]: Communication line voltage

2. UART0 and UART3 cannot communicate at different potential. Use UART1 for communication at different potential.

(2) Serial interface: Serial array unit (10/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(f) Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{SCK10}$... internal clock output) (1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ	500			ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1000			ns
$\overline{SCK10}$ high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ	t_{KCY1}/2 – 120			ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 275			ns
$\overline{SCK10}$ low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 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 = 50 pF, R_b = 2.7 kΩ	t_{KCY1}/2 – 35			ns
SI10 setup time (to $\overline{SCK10}$↑) Note	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ	195			ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	380			ns
SI10 hold time (from $\overline{SCK10}$↑) Note	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 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 = 50 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{SCK10}$↓ to SO10 output Note	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ			165	ns
		2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ			320	ns

Note When DAP02 = 0 and CKP02 = 0, or DAP02 = 1 and CKP02 = 1.

Caution Select the TTL input buffer for SI10 and the N-ch open drain output (V_{DD} tolerance) mode for SO10 and $\overline{SCK10}$ by using the PIM0 and POM0 registers.

Remarks 1. R_b[Ω]:Communication line ($\overline{SCK10}$, SO10) pull-up resistance,
C_b[F]: Communication line (SI10, SO10, $\overline{SCK10}$) load capacitance, V_b[V]: Communication line voltage

2. 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

3. CSI00 cannot communicate at different potential. Use CSI10 for communication at different potential.

(2) Serial interface: Serial array unit (11/17)

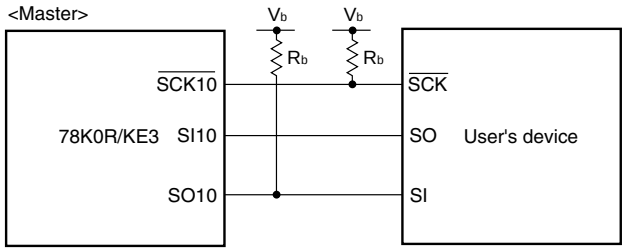
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(f) Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{SCK10}$... internal clock output) (2/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SI10 setup time (to $\overline{SCK10}$↓) Note	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 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 = 50 pF, R_b = 2.7 kΩ	100			ns
SI10 hold time (from $\overline{SCK10}$↓) Note	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 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 = 50 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{SCK10}$↑ to SO10 output Note	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 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 = 50 pF, R_b = 2.7 kΩ			40	ns

Note When DAP02 = 0 and CKP02 = 1, or DAP02 = 1 and CKP02 = 0.

CSI mode connection diagram (communication at different potential)



Caution Select the TTL input buffer for SI10 and the N-ch open drain output (V_{DD} tolerance) mode for SO10 and $\overline{SCK10}$ by using the PIM0 and POM0 registers.

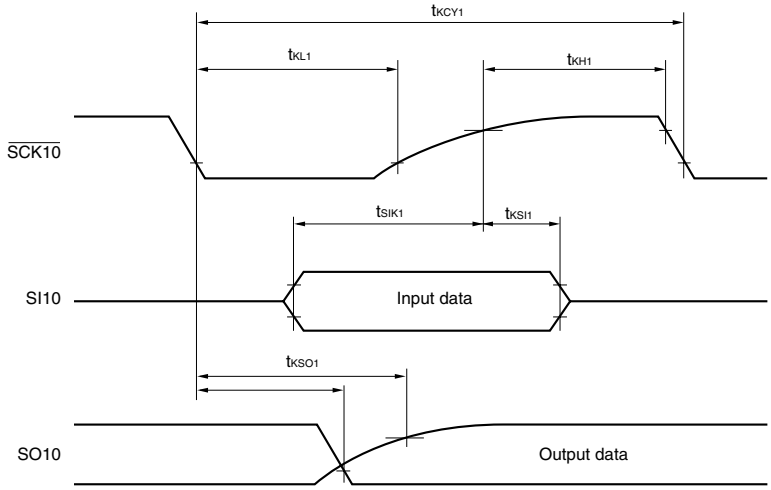
Remarks 1. R_b[Ω]:Communication line ($\overline{SCK10}$, SO10) pull-up resistance,
C_b[F]: Communication line (SI10, SO10, $\overline{SCK10}$) load capacitance, V_b[V]: Communication line voltage

2. 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

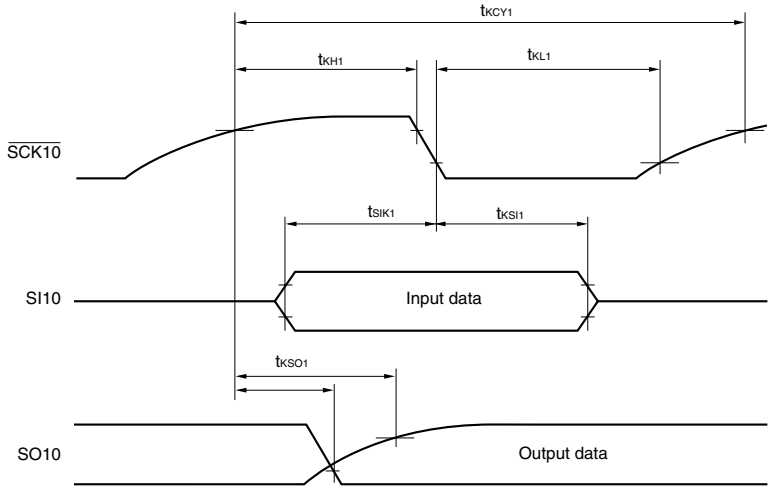
3. CSI00 cannot communicate at different potential. Use CSI10 for communication at different potential.

(2) Serial interface: Serial array unit (12/17)

CSI mode serial transfer timing (communication at different potential)
(When DAP02 = 0 and CKP02 = 0, or DAP02 = 1 and CKP02 = 1.)



CSI mode serial transfer timing (communication at different potential)
(When DAP02 = 0 and CKP02 = 1, or DAP02 = 1 and CKP02 = 0.)



Caution Select the TTL input buffer for SI10 and the N-ch open drain output (V_{DD} tolerance) mode for SO10 and $\overline{\text{SCK10}}$ by using the PIM0 and POM0 registers.

Remark CSI00 cannot communicate at different potential. Use CSI10 for communication at different potential.

(2) Serial interface: Serial array unit (13/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(g) Communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, $\overline{SCK10}$... external clock input)

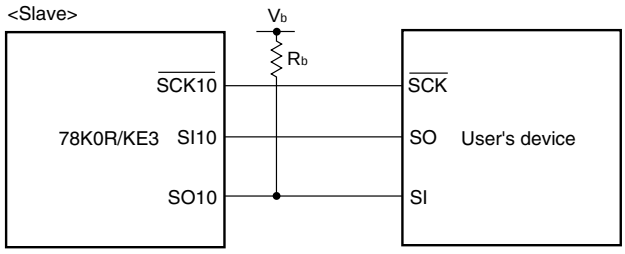
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCK10}$ cycle time	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	16.6 MHz < f_{MCK}	12/f_{MCK}			ns
			12.5 MHz < f_{MCK} ≤ 16.6 MHz	10/f_{MCK}			ns
			8.3 MHz < f_{MCK} ≤ 12.5 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 8.3 MHz	6/f_{MCK}			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	17.5 MHz < f_{MCK}	18/f_{MCK}			ns
			15 MHz < f_{MCK} ≤ 17.5 MHz	16/f_{MCK}			ns
			12.5 MHz < f_{MCK} ≤ 15 MHz	14/f_{MCK}			ns
			10 MHz < f_{MCK} ≤ 12.5 MHz	12/f_{MCK}			ns
			7.5 MHz < f_{MCK} ≤ 10 MHz	10/f_{MCK}			ns
			5 MHz < f_{MCK} ≤ 7.5 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 5 MHz	6/f_{MCK}			ns
$\overline{SCK10}$ 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
SI10 setup time (to $\overline{SCK10}$↑) Note 1	t_{SIK2}			1/f_{MCK} + 90			ns
SI10 hold time (from $\overline{SCK10}$↑) Note 2	t_{KSI2}			50			ns
Delay time from $\overline{SCK10}$↓ to SO10 output Note 3	t_{KSO2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 1.4 kΩ				1/f_{MCK} + 245	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ				1/f_{MCK} + 400	ns

Notes 1. When DAP02 = 0 and CKP02 = 0, or DAP02 = 1 and CKP02 = 1. The SI10 setup time becomes "to $\overline{SCK10}$↓" when DAP02 = 0 and CKP02 = 1, or DAP02 = 1 and CKP02 = 0.

2. When DAP02 = 0 and CKP02 = 0, or DAP02 = 1 and CKP02 = 1. The SI10 hold time becomes "from $\overline{SCK10}$↓" when DAP02 = 0 and CKP02 = 1, or DAP02 = 1 and CKP02 = 0.

3. When DAP02 = 0 and CKP02 = 0, or DAP02 = 1 and CKP02 = 1. The delay time to SO10 output becomes "from $\overline{SCK10}$↑" when DAP02 = 0 and CKP02 = 1, or DAP02 = 1 and CKP02 = 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 SI10 and $\overline{SCK10}$ and the N-ch open drain output (V_{DD} tolerance) mode for SO10 by using the PIM0 and POM0 registers.

Remarks 1. R_b[Ω]:Communication line (SO10) pull-up resistance,
C_b[F]: Communication line (SO10, $\overline{SCK10}$) load capacitance, V_b[V]: Communication line voltage

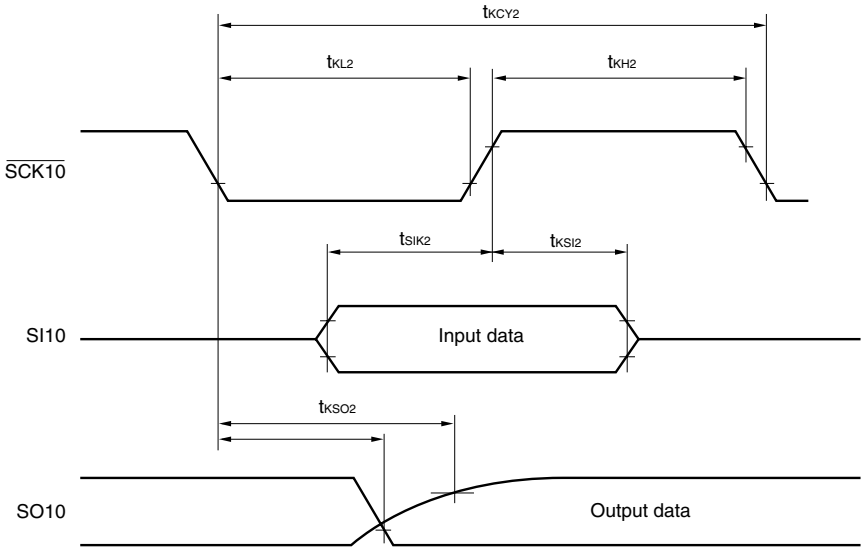
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS02 bit of the SMR02 register.)

3. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.
$4.0\ V \le V_{DD} \le 5.5\ V$, $2.7\ V \le V_b \le 4.0\ V$: $V_{IH} = 2.2\ V$, $V_{IL} = 0.8\ V$
$2.7\ V \le V_{DD} \le 4.0\ V$, $2.3\ V \le V_b \le 2.7\ V$: $V_{IH} = 2.0\ V$, $V_{IL} = 0.5\ V$

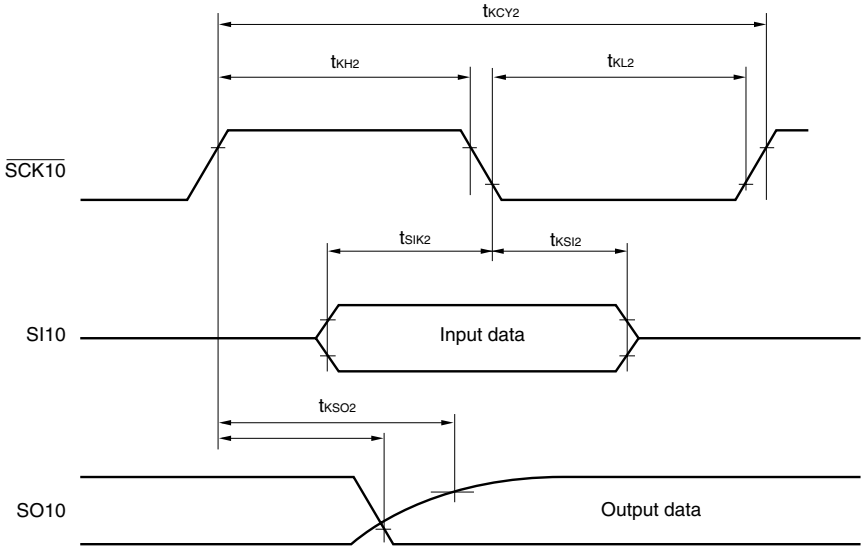
4. CSI00 cannot communicate at different potential. Use CSI10 for communication at different potential.

(2) Serial interface: Serial array unit (15/17)

CSI mode serial transfer timing (communication at different potential)
(When DAP02 = 0 and CKP02 = 0, or DAP02 = 1 and CKP02 = 1.)



CSI mode serial transfer timing (communication at different potential)
(When DAP02 = 0 and CKP02 = 1, or DAP02 = 1 and CKP02 = 0.)



Caution Select the TTL input buffer for SI10 and $\overline{\text{SCK10}}$ and the N-ch open drain output (V_{DD} tolerance) mode for SO10 by using the PIM0 and POM0 registers.

Remark CSI00 cannot communicate at different potential. Use CSI10 for communication at different potential.

(2) Serial interface: Serial array unit (16/17)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(h) Communication at different potential (2.5 V, 3 V) (simplified I^2C mode)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCL10 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	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	kHz
Hold time when SCL10 = “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	1065		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	1065		ns
Hold time when SCL10 = “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	445		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	445		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	160	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	160	ns

Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for SDA10 and the N-ch open drain output (V_{DD} tolerance) mode for SCL10 by using the PIM0 and POM0 registers.

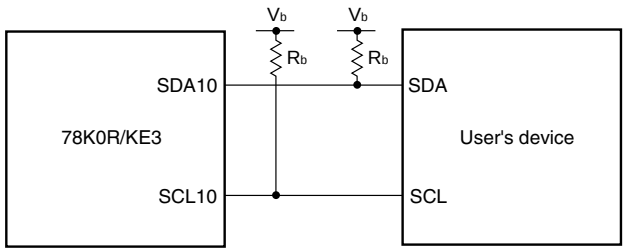
Remarks 1. R_b[Ω]:Communication line (SDA10, SCL10) pull-up resistance,
C_b[F]: Communication line (SDA10, SCL10) load capacitance, V_b[V]: Communication line voltage

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS02 bit of the SMR02 register.)

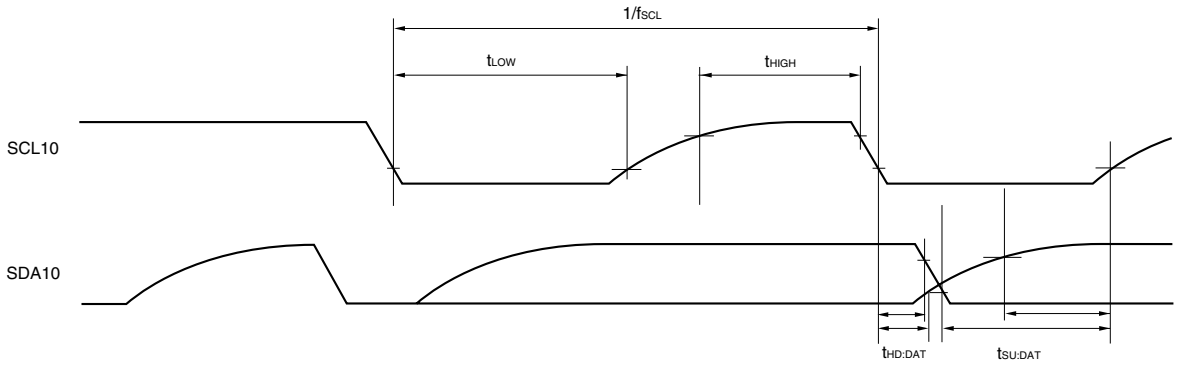
3. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in simplified I^2C mode mode.
4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} ≤ 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(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 SDA10 and the N-ch open drain output (V_{DD} tolerance) mode for SCL10 by using the PIM0 and POM0 registers.

Remark R_b[Ω]:Communication line (SDA10, SCL10) pull-up resistance, V_b[V]: Communication line voltage

(3) Serial interface: IIC0

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(a) IIC0

Parameter	Symbol	Conditions	Standard Mode		High-Speed Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCL0 clock frequency	f_{SCL}	6.7 MHz ≤ f_{CLK}	0	100	0	400	kHz
		4.0 MHz ≤ f_{CLK} < 6.7 MHz	0	100	0	340	kHz
		3.2 MHz ≤ f_{CLK} < 4.0 MHz	0	100	–	–	kHz
		2.0 MHz ≤ f_{CLK} < 3.2 MHz	0	85	–	–	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]	$t_{HD:DAT}$	CL00 = 1 and CL01 = 1	0	3.45[Note 3]	0	0.9[Note 4]	μs
				5.50[Note 5]		1.5[Note 6]	μs
		CL00 = 0 and CL01 = 0, or CL00 = 1 and CL01 = 0	0	3.45	0	0.9[Note 7]	μs
						0.95[Note 8]	μs
		CL00 = 0 and CL01 = 1	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.

3. When 3.2 MHz ≤ f_{CLK} ≤ 4.19 MHz.

4. When 6.7 MHz ≤ f_{CLK} ≤ 8.38 MHz.

5. When 2.0 MHz ≤ f_{CLK} < 3.2 MHz. At this time, use the SCL0 clock within 85 kHz.

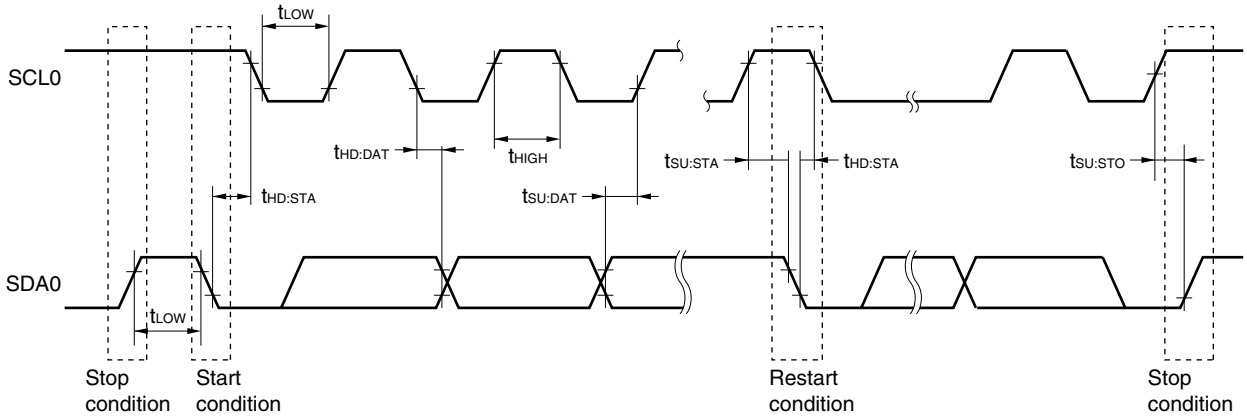
6. When 4.0 MHz ≤ f_{CLK} < 6.7 MHz. At this time, use the SCL0 clock within 340 kHz.

7. When 8.0 MHz ≤ f_{CLK} ≤ 16.76 MHz.

8. When 7.6 MHz ≤ f_{CLK} < 8.0 MHz.

Remark CL00, CL01, DFC0: Bits 0, 1, and 2 of the IIC clock select register 0 (IICCL0)

IIC0 serial transfer timing



(4) Serial interface: On-chip debug (UART)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 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			2.66	Mbps
TOOL1 output frequency	f_{TOOL1}	2.7 V ≤ V_{DD} ≤ 5.5 V			10	MHz
		1.8 V ≤ V_{DD} < 2.7 V			2.5	MHz

A/D Converter Characteristics

(T_A = –40 to +85°C, 2.3 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.3 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}				10	bit
Overall error[Notes 1, 2]	AINL	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
		2.3 V ≤ AV_{REF} < 2.7 V			±1.2	%FSR
Conversion time	t_{CONV}	4.0 V ≤ AV_{REF} ≤ 5.5 V	6.1		66.6	μs
		2.7 V ≤ AV_{REF} < 4.0 V	12.2		66.6	μs
		2.3 V ≤ AV_{REF} < 2.7 V	27		66.6	μs
Zero-scale error[Notes 1, 2]	EZS	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
		2.3 V ≤ AV_{REF} < 2.7 V			±0.6	%FSR
Full-scale error[Notes 1, 2]	EFS	4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
		2.7 V ≤ AV_{REF} < 4.0 V			±0.6	%FSR
		2.3 V ≤ AV_{REF} < 2.7 V			±0.6	%FSR
Integral non-linearity error[Note 1]	ILE	4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±4.5	LSB
		2.3 V ≤ AV_{REF} < 2.7 V			±6.5	LSB
Differential non-linearity error[Note 1]	DLE	4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
		2.7 V ≤ AV_{REF} < 4.0 V			±2.0	LSB
		2.3 V ≤ AV_{REF} < 2.7 V			±2.0	LSB
Analog input voltage	V_{AIN}	2.3 V ≤ AV_{REF} ≤ 5.5 V	AV_{SS}		AV_{REF}	V

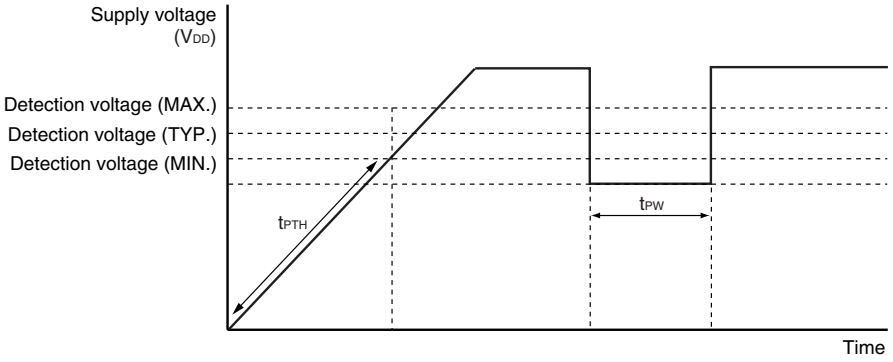
Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

POC Circuit Characteristics (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC0}		1.5	1.59	1.68	V
Power supply voltage rise inclination	t_{PTH}	Change inclination of V_{DD}: 0 V → V_{POC0}	0.5			V/ms
Minimum pulse width	t_{PW}	When the voltage drops	200			μs
Detection delay time					200	μs

POC Circuit Timing



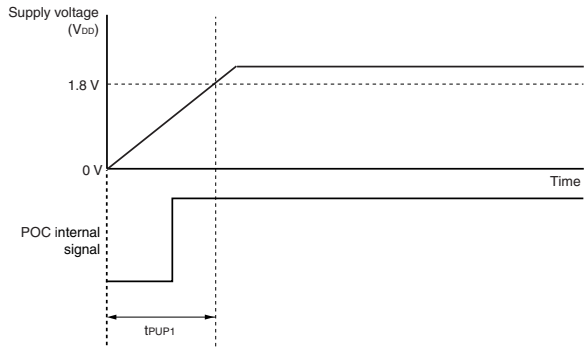
Supply Voltage Rise Time (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 1.8 V (V_{DD} (MIN.))[Note] (V_{DD}: 0 V → 1.8 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 1.8 V (V_{DD} (MIN.))[Note] (releasing $\overline{\text{RESET}}$ input → V_{DD}: 1.8 V)	t_{PUP2}	LVI default start function stopped is set (LVIOFF (Option Byte) = 1), when $\overline{\text{RESET}}$ input is used			1.88	ms

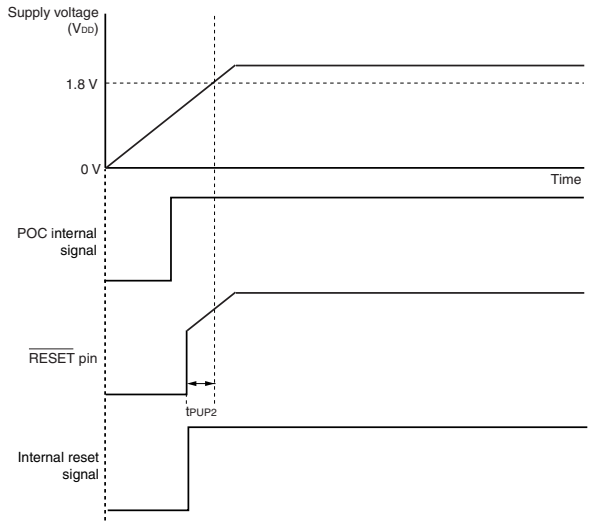
Note Make sure to raise the power supply in a shorter time than this.

Supply Voltage Rise Time Timing

<R> • 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, $V_{POC} \le V_{DD} = EV_{DD} \le 5.5$ V, $V_{SS} = EV_{SS}$ =0 V)

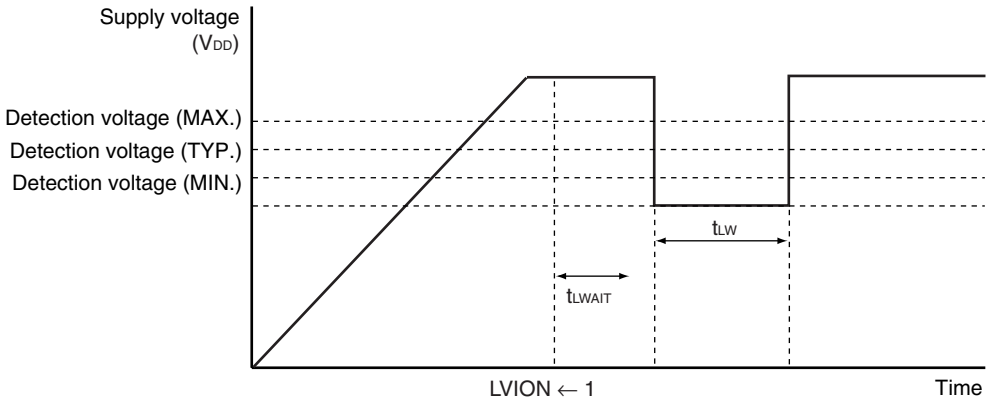
Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.12	4.22	4.32	V
		V_{LVI1}		3.97	4.07	4.17	V
		V_{LVI2}		3.82	3.92	4.02	V
		V_{LVI3}		3.66	3.76	3.86	V
		V_{LVI4}		3.51	3.61	3.71	V
		V_{LVI5}		3.35	3.45	3.55	V
		V_{LVI6}		3.20	3.30	3.40	V
		V_{LVI7}		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
		V_{LVI10}		2.58	2.68	2.78	V
		V_{LVI11}		2.43	2.53	2.63	V
		V_{LVI12}		2.28	2.38	2.48	V
		V_{LVI13}		2.12	2.22	2.32	V
		V_{LVI14}		1.97	2.07	2.17	V
		V_{LVI15}		1.81	1.91	2.01	V
	External input pin[Note 1]	V_{EXLVI}	EXLVI < V_{DD}, 1.8 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						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

Remark $V_{LVI(n-1)} > V_{LVIn}$: n = 1 to 15

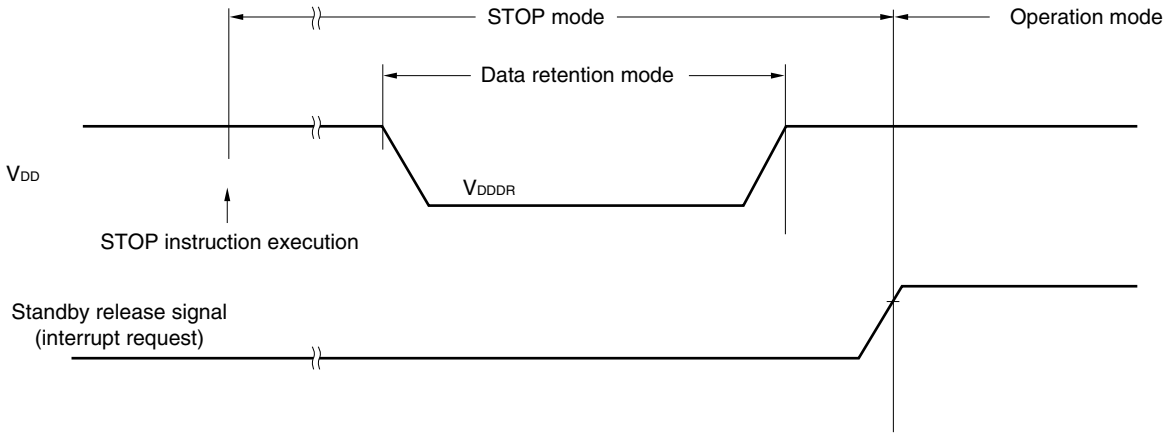
LVI Circuit Timing



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 until a POC reset is effected, but data is not retained when a POC reset is effected.



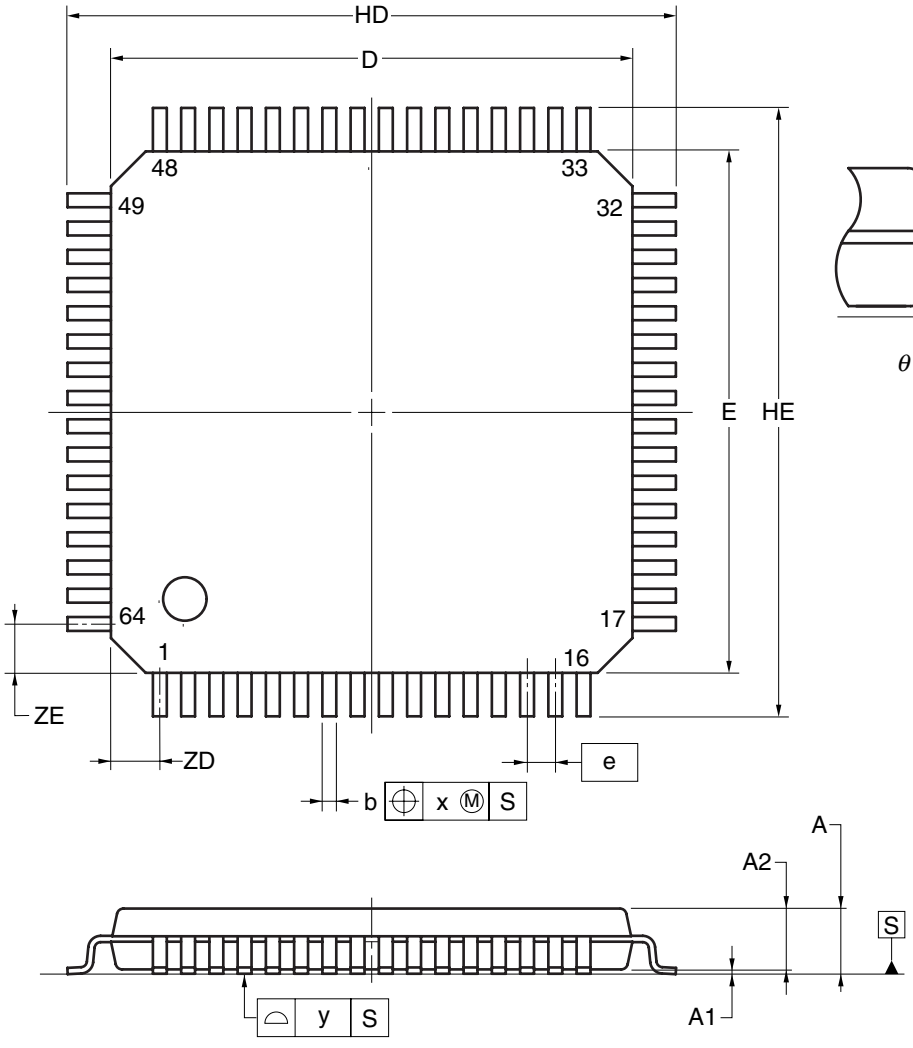
Flash Memory Programming Characteristics

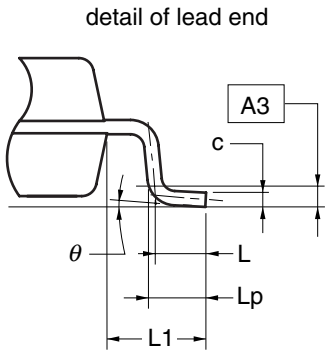
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

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 per chip	C_{erwr}	Retention: 15 years 1 erase + 1 write after erase = 1 rewrite[Note]	100			Times

Note When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

64-PIN PLASTIC LQFP (12x12)





(UNIT:mm)

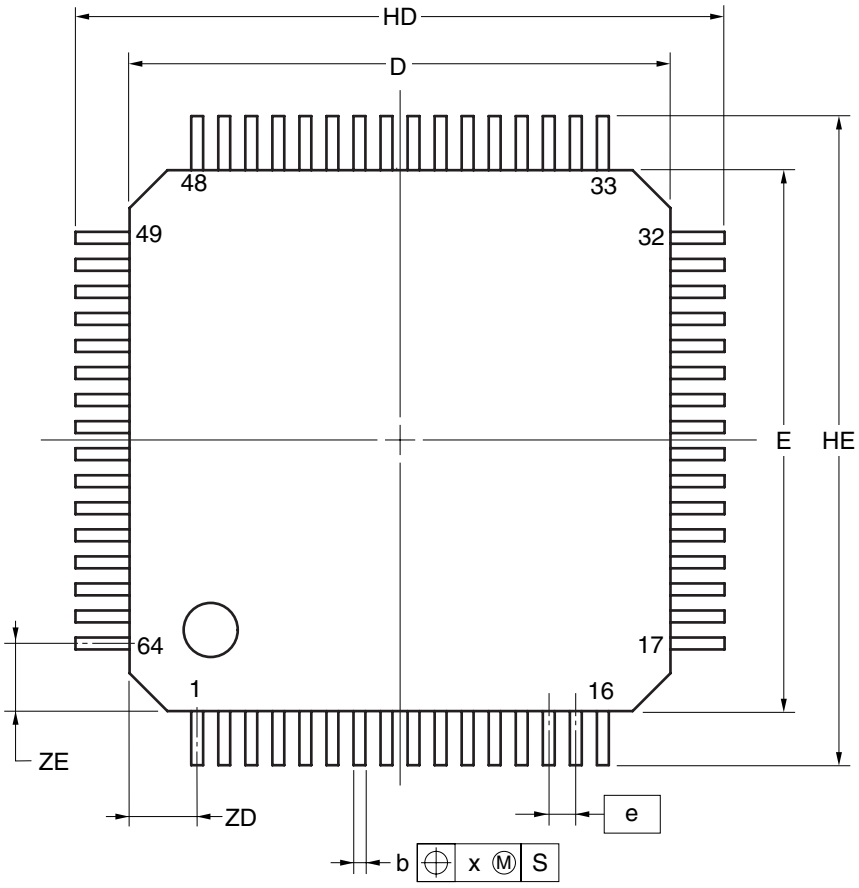
ITEM	DIMENSIONS
D	12.00±0.20
E	12.00±0.20
HD	14.00±0.20
HE	14.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.30^{+0.08}_{-0.04}$
c	$0.125^{+0.75}_{-0.25}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.65
x	0.13
y	0.10
ZD	1.125
ZE	1.125

NOTE
Each lead centerline is located within 0.13 mm of its true position at maximum material condition.

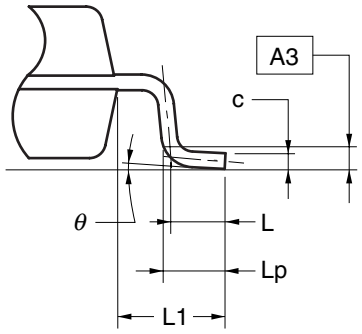
P64GK-65-GAJ

© NEC Electronics Corporation 2005

64-PIN PLASTIC LQFP(FINE PITCH)(10x10)



detail of lead end



(UNIT:mm)

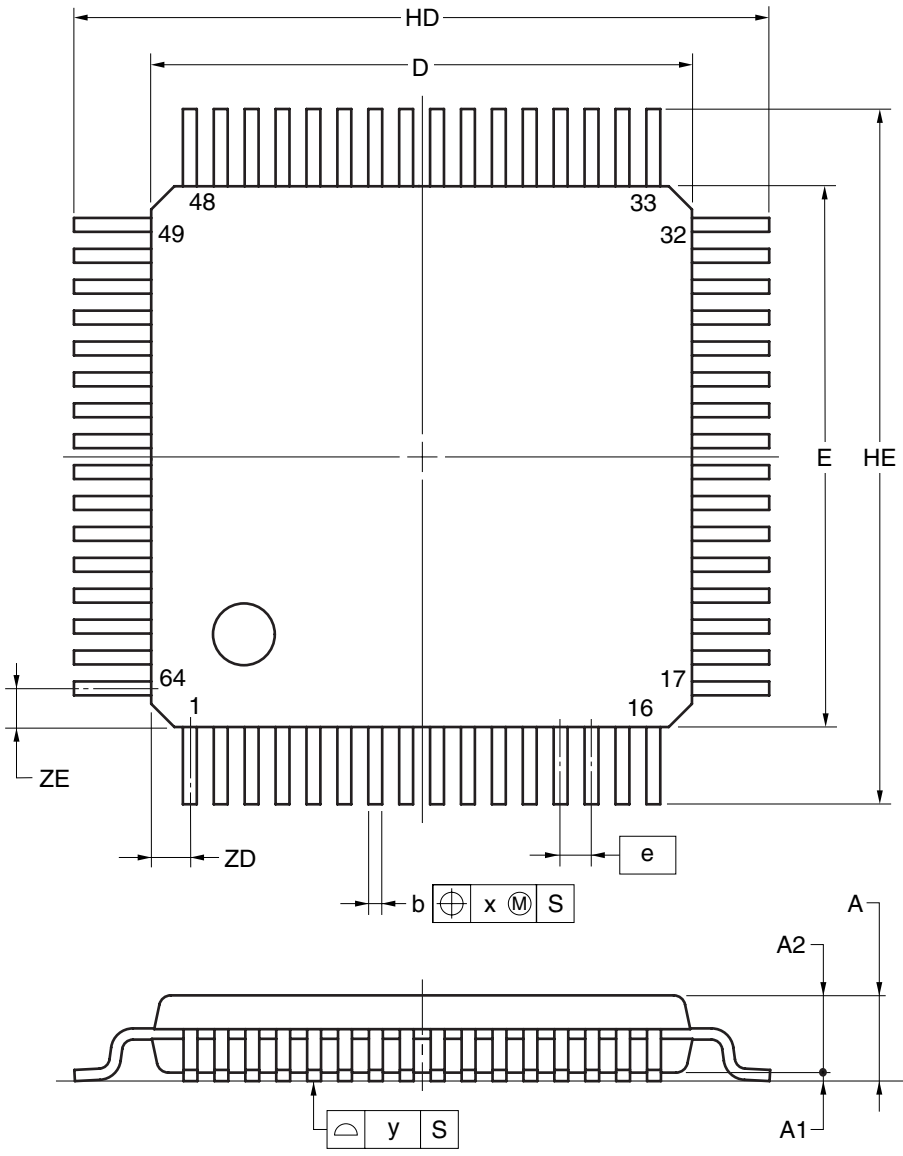
ITEM	DIMENSIONS
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

P64GB-50-GAH

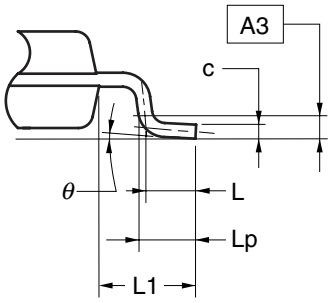
NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

© NEC Electronics Corporation 2005

<R> 64-PIN PLASTIC TQFP (FINE PITCH) (7x7)



detail of lead end



(UNIT:mm)

ITEM	DIMENSIONS
D	7.00±0.20
E	7.00±0.20
HD	9.00±0.20
HE	9.00±0.20
A	1.20 MAX.
A1	0.10±0.05
A2	1.00±0.05
A3	0.25
b	$0.16^{+0.07}_{-0.03}$
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	0.40
x	0.07
y	0.08
ZD	0.50
ZE	0.50

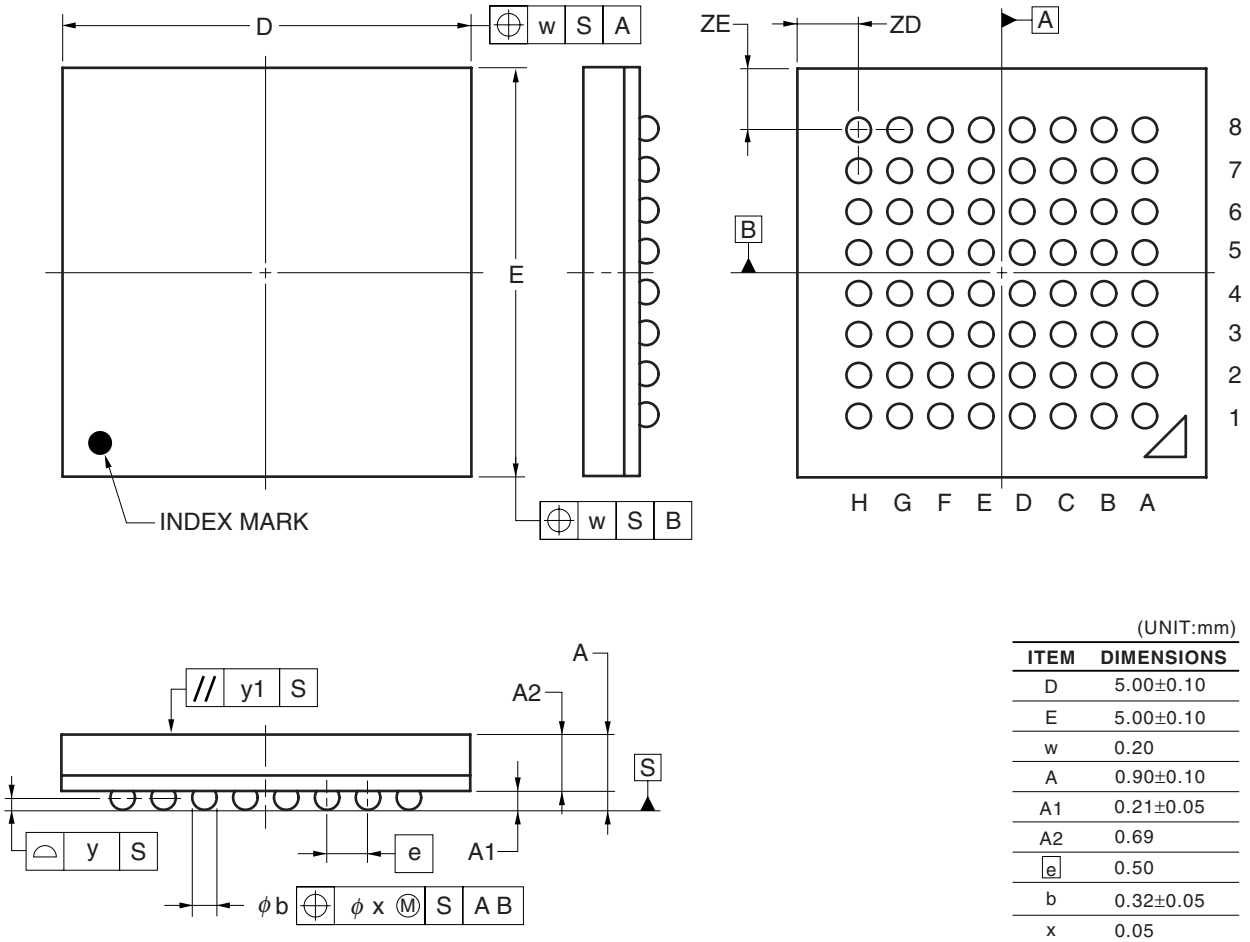
P64GA-40-HAB

NOTE
Each lead centerline is located within 0.07mm of its true position at maximum material condition.

© NEC Electronics Corporation 2005

<R>

64-PIN PLASTIC FBGA (5x5)



(UNIT:mm)

ITEM	DIMENSIONS
D	5.00±0.10
E	5.00±0.10
w	0.20
A	0.90±0.10
A1	0.21±0.05
A2	0.69
e	0.50
b	0.32±0.05
x	0.05
y	0.08
y1	0.20
ZD	0.75
ZE	0.75

P64F1-50-AN1

© NEC Electronics Corporation 2006

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the 78K0R/KE3.
Figure A-1 shows the development tool configuration.

- **Support for PC98-NX series**

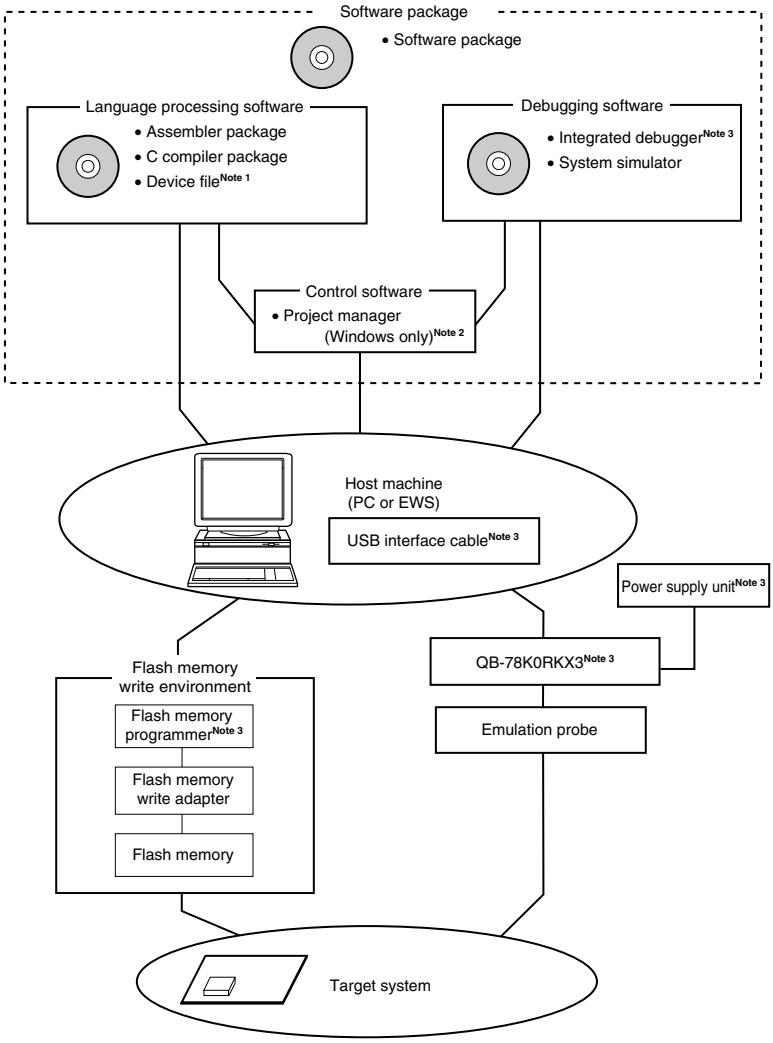
  Unless otherwise specified, products supported by IBM PC/AT™ compatibles are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT compatibles.

- **Windows™**

  Unless otherwise specified, “Windows” means the following OSs.
  - Windows 98
  - Windows NT™
  - Windows 2000
  - Windows XP

Figure A-1. Development Tool Configuration (1/2)

(1) When using the in-circuit emulator QB-78K0RKX3



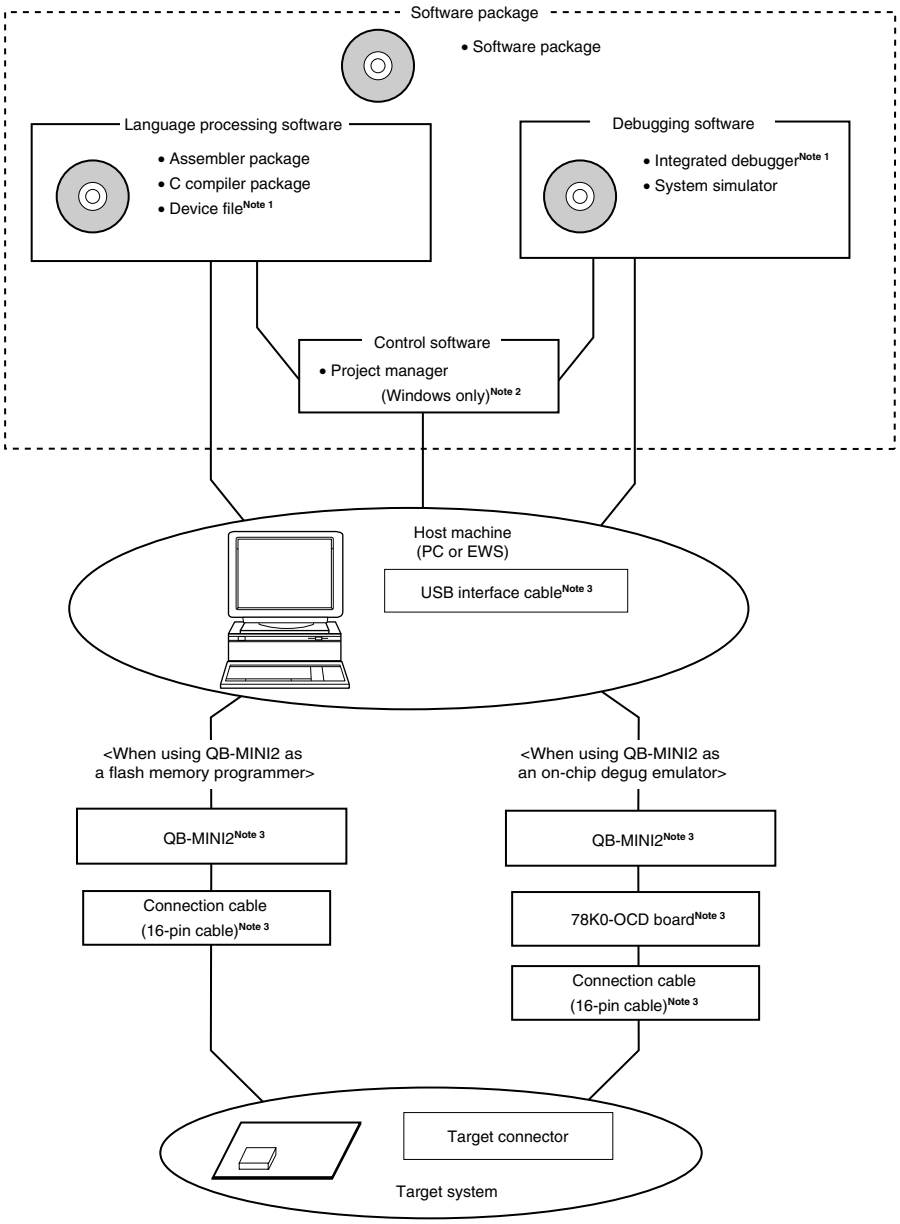
Notes 1. Download the device file for 78K0R/KE3 (DF781188) from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

2. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

3. In-circuit emulator QB-78K0RKX3 is supplied with integrated debugger ID78K0R-QB, on-chip debug emulator with programming function QB-MINI2, power supply unit, and USB interface cable. Any other products are sold separately.

Figure A-1. Development Tool Configuration (2/2)

(2) When using the on-chip debug emulator with programming function QB-MINI2



Notes 1. Download the device file for 78K0R/KE3 (DF781188) and the integrated debugger (ID78K0R-QB) from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

2. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

3. On-chip debug emulator QB-MINI2 is supplied with USB interface cable, connection cables (10-pin cable and 16-pin cable), and 78K0-OCD board. Any other products are sold separately. In addition, download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html).

A.1 Software Package

SP78K0R 78K0R Series software package	Development tools (software) common to the 78K0R microcontrollers are combined in this package.
	Part number: 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 (DF781188) (sold separately). **<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
DF781188[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××××DF781188

Note The DF781188 can be used in common with the RA78K0R, CC78K0R, SM+ for 78K0R, and ID78K0R-QB. Download the DF781188 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

Remark ××××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××××DF781188

××××	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

<R> A.4.1 When using flash memory programmer FG-FP5, FL-PR5, FG-FP4, and FL-PR4

FL-PR4, PG-FP4, FL-PR5, PG- FP5 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
FA-78F1146GK-GAJ-RX (RoHS supported), FA-78F1146GB-GAH-RX (RoHS supported), FA-78F1146GA-HAB-RX (RoHS supported) Note, FA-78F1146F1-AN1-RX (RoHS supported) Note Flash memory programming adapter	Flash memory programming adapter used connected to the flash memory programmer for use. • FA-78F1146GK-GAJ-RX: 64-pin plastic LQFP (GK-GAJ type) • FA-78F1146GB-GAH-RX: 64-pin plastic LQFP (GB-GAH type) • FA-78F1146GA-HAB-RX: 64-pin plastic TQFP (GA-HAB type) Note • FA-78F1146F1-AN1-RX: 64-pin plastic FBGA (F1-AN1 type) Note

Note Under development

Remark The FL-PR4, FL-PR5, FA-78F1146GK-GAJ-RX, FA-78F1146GB-GAH-RX, FA-78F1146GA-HAB-RX, and FA-78F1146F1-AN1-RX are a product of Naito Densei Machida Mfg. Co., Ltd.

<R>

A.4.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This is a flash memory programmer dedicated to microcontrollers with on-chip flash memory. It is available also as on-chip debug emulator which serves to debug hardware and software when developing application systems using the 78K0R. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. To use 78K0R/KE3, use USB interface cable and 16-pin connection cable.

Remark Download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html).

A.5 Debugging Tools (Hardware)

A.5.1 When using in-circuit emulator QB-78K0RKX3

QB-78K0KX3 In-circuit emulator	This in-circuit emulator serves to debug hardware and software when developing application systems using the 78K0R/Kx3. It supports to the integrated debugger (ID78K0R-QB). This emulator should be used in combination with a power supply unit and emulation probe, and the USB is used to connect this emulator to the host machine.
QB-144-CA-01 Check pin adapter	This check pin adapter is used in waveform monitoring using the oscilloscope, etc.
QB-144-EP-02S Emulation probe	This emulation probe is flexible type and used to connect the in-circuit emulator and target system.
<R> QB-64GK-EA-06T, QB-64GB-EA-08T, QB-64GA-EA-02T[Note 1], QB-64F1-EA-01T[Note 1] Exchange adapter	This exchange adapter is used to perform pin conversion from the in-circuit emulator to target connector. • QB-64GK-EA-06T: 64-pin plastic LQFP (GK-GAJ type) • QB-64GB-EA-08T: 64-pin plastic LQFP (GB-GAH type) • QB-64GA-EA-02T: 64-pin plastic TQFP (GA-HAB type)[Note 1] • QB-64F1-EA-01T: 64-pin plastic FBGA (F1-AN1 type)[Note 1]
<R> QB-64GK-YS-01T, QB-64GB-YS-01T, QB-64GA-YS-01T[Note 1] Space adapter[Note 2]	This space adapter is used to adjust the height between the target system and in-circuit emulator. • QB-64GK-YS-01T: 64-pin plastic LQFP (GK-GAJ type) • QB-64GB-YS-01T: 64-pin plastic LQFP (GB-GAH type) • QB-64GA-YS-01T: 64-pin plastic TQFP (GA-HAB type)[Note 1]
<R> QB-64GK-YQ-01T, QB-64GB-YQ-01T, QB-64GA-YQ-01T[Note 1] YQ connector[Note 2]	This YQ connector is used to connect the target connector and exchange adapter. • QB-64GK-YQ-01T: 64-pin plastic LQFP (GK-GAJ type) • QB-64GB-YQ-01T: 64-pin plastic LQFP (GB-GAH type) • QB-64GA-YQ-01T: 64-pin plastic TQFP (GA-HAB type)[Note 1]
<R> QB-64GK-HQ-01T, QB-64GB-HQ-01T, QB-64GA-HQ-01T[Note 1] Mount adapter[Note 2]	This mount adapter is used to mount the target device with socket. • QB-64GK-HQ-01T: 64-pin plastic LQFP (GK-GAJ type) • QB-64GB-HQ-01T: 64-pin plastic LQFP (GB-GAH type) • QB-64GA-HQ-01T: 64-pin plastic TQFP (GA-GAB type)[Note 1]
<R> QB-64GK-NQ-01T, QB-64GB-NQ-01T, QB-64GA-NQ-01T[Note 1], QB-64FC-NQ-01T[Note 1] Target connector	This target connector is used to mount on the target system. • QB-64GK-NQ-01T: 64-pin plastic LQFP (GK-GAJ type) • QB-64GB-NQ-01T: 64-pin plastic LQFP (GB-GAH type) • QB-64GA-NQ-01T: 64-pin plastic TQFP (GA-HAB type)[Note 1] • QB-64FC-NQ-01T: 64-pin plastic FBGA (F1-AN1 type)[Note 1]

Notes 1. Under development

2. These adapter are not necessary in 64-pin plastic FBGA (F1-AN1 type).

Remarks 1. The QB-78K0RKX3 is supplied with a power supply unit and USB interface cable. As control software, integrated debugger ID78K0R-QB and on-chip debug emulator with programming function QB-MINI2 are supplied.

2. The packed contents differ depending on the part number, as follows.

Packed Contents / Part Number	In-Circuit Emulator	Emulation Probe	Exchange Adapter	YQ Connector	Target Connector
QB-78K0RKX3-ZZZ	QB-78K0RKX3	None			
QB-78K0RKX3-T64GK		QB-144-EP-02S	QB-64GK-EA-06T	QB-64GK-YQ-01T	QB-64GK-NQ-01T
QB-78K0RKX3-T64GB			QB-64GB-EA-08T	QB-64GB-YQ-01T	QB-64GB-NQ-01T
<R> QB-78K0RKX3-T64GA[Note]			QB-64GA-EA-02T[Note]	QB-64GA-YQ-01T[Note]	QB-64GA-NQ-01T[Note]
<R> QB-78K0RKX3-T64F1[Note]			QB-64F1-EA-01T[Note]	None	QB-64FC-NQ-01T[Note]

Note Under development

<R> 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. It is available also as flash memory programmer dedicated to microcontrollers with on-chip flash memory. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. To use 78K0R/KE3, use USB interface cable and 16-pin connection cable.

Remark Download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html).

A.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 (DF781188) (sold separately).
	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 (sold separately).
	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

(1/4)

Page	Description	Classification
Throughout	Change of status of μPD78F1142, 78F1143, 78F1144, 78F1145, and 78F1146 from under development to mass production	(d)
CHAPTER 1 OUTLINE		
p.17	Addition of package and **Note** to **1.3 Ordering Information**	(d)
pp.18, 19	Addition of package and **Note** to **1.4 Pin Configuration (Top View)**	(d)
p.24	Change of **1.7 Outline of Functions**	(d)
CHAPTER 2 PIN FUNCTIONS		
p.25	Change of corresponding pins of EV_{DD} and V_{DD} in **Table 2-1. Pin I/O Buffer Power Supplies**	(c)
p.37	Change of description in **2.2.15 REGC**	(b)
p.37	Change of description in **2.2.18 FLMD0**	(c)
p.39	Modification of 37-A to 37-B and 39 to 2-W in **Table 2-2. Connection of Unused Pins**	(c)
pp.40, 41	Modification of 37-A to 37-B and 39 to 2-W in **Figure 2-1. Pin I/O Circuit List**	(c)
CHAPTER 3 CPU ARCHITECTURE		
p.62	Change of address in **Figure 3-16. Configuration of General-Purpose Registers**	(a)
pp.67, 68	Addition of register and **Note** in **Table 3-5. SFR List**	(c)
CHAPTER 4 PORT FUNCTIONS		
Throughout	Addition of PIM register and POM register in block diagram	(c)
p.88	Change of corresponding pins of EV_{DD} and V_{DD} in **Table 4-1. Pin I/O Buffer Power Supplies**	(c)
p.92	Change of **Cautions 1** and **Cautions 2** in **4.2.1 Port 0**	(c)
p.98	Change of **Cautions 1, Cautions 2, and Cautions 3** in **4.2.2 Port 1**	(c)
p.106	Change of **Cautions 1** and addition of **Cautions 2** in **4.2.4 Port 3**	(c)
p.107	Change of **Cautions 2** in **4.2.5 Port 4**	(c)
p.114	Addition of **Caution** to **4.2.7 Port 6**	(c)
p.121	Addition of **Caution** to **4.2.11 Port 14**	(c)
p.128	Addition description to **(4) Port input mode registers (PIM0)** and **(5) Port output mode registers (POM0)** in **4.3**	(c)
p.128	Change of **Figure 4-32. Format of Port Input Mode Register**	(a)
CHAPTER 5 CLOCK GENERATOR		
p.146	Addition of **Notes 3** to **Figure 5-6 Format of System Clock Control Register (CKC)**	(c)
p.150	Addition of **Cautions 5** to **Figure 5-8. Format of Operation Speed Mode Control Register (OSMC)**	(b)
CHAPTER 6 TIMER ARRAY UNIT		
p.180	Change of **Table 6-1. Configuration of Timer Array Unit**	(a)
pp.180, 201, 217, 218, 220, 224, 225, 228, 232	Deletion of bit 7 (TOM07) of TOM0 register	(a)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(2/4)

Page	Description	Classification
CHAPTER 6 TIMER ARRAY UNIT (continuation)		
p.187	Change of description of MASTER0n bit in **Figure 6-6. Format of Timer Mode Register 0n (TMR0n) (1/3)**	(c)
p.197	Change of **Figure 6-16. Format of Timer Input Select Register 0 (TIS0)** and **Caution**	(c)
p.199	Addition of description to **6.3 (10) Timer output register 0 (TO0)**	(c)
p.201	Addition of description to **6.3 (12) Timer output mode register 0 (TOM0)**	(c)
p.201	Change of Remark in **Figure 6-20. Format of Timer Output Mode Register 0 (TOM0)**	(c)
p.202	Change of Remark in **Figure 6-21. Format of Input Switch Control Register (ISC)**	(c)
CHAPTER 7 REAL-TIME COUNTER		
p.257	Change of **Cautions 1** in **Figure 7-2. Format of Peripheral Enable Register 0 (PER0)**	(c)
p.267	Addition of description to **7.3 (15) Alarm hour register (ALARMWH)**	(c)
p.269	Addition of **Note** to Figure 7-18. Procedure for Starting Operation of Real-Time Counter	(c)
CHAPTER 8 WATCHDOG TIMER		
p.275	Change of **Cautions 1** and **Cautions 2** in **8.3 (1) Watchdog timer enable register (WDTE)**	(a)
CHAPTER 11 SERIAL ARRAY UNIT		
Throughout	Change of SOm register	(a)
p.310	Change of **Figure 11-1. Block Diagram of Serial Array Unit 0**	(a)
p.311	Change of **Figure 11-2. Block Diagram of Serial Array Unit 1**	(a)
p.330	Change of description in **11.3 (12) Serial output register m (SOm)**	(c)
pp.336 to 338	Addition of **11.4 Operation stop mode**	(c)
p.344	Change of **Figure 11-27. Procedure for Resuming Master Transmission**	(c)
p.353	Change of **Figure 11-36. Timing Chart of Master Reception (in Single-Reception Mode)**	(a)
p.358	Change of **Figure 11-41. Procedure for Resuming Master Transmission/Reception**	(a)
p.359	Change of **Figure 11-42. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode)**	(a)
p.361	Change of **Figure 11-44. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**	(a)
p.362	Change of **Figure 11-45. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**	(a)
p.367	Change of **Figure 11-49. Procedure for Resuming Slave Transmission**	(a)
p.368	Change of **Figure 11-50. Timing Chart of Slave Transmission (in Single-Transmission Mode)**	(c)
p.375	Change of **Figure 11-57. Procedure for Resuming Slave Reception**	(a)
p.376	Change of **Figure 11-58. Timing Chart of Slave Reception (in Single-Reception Mode)**	(a)
p.382	Change of **Figure 11-63. Procedure for Resuming Slave Transmission/Reception**	(a)
p.383	Change of **Figure 11-64. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)**	(a)
p.385	Change of **Figure 11-66. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**	(a)
p.386	Change of **Figure 11-67. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**	(a)
p.400	Change of Transfer data length in **11.6.2 UART reception**	(a)
p.405	Change of **Figure 11-80. Timing Chart of UART Reception**	(a)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(3/4)

Page	Description	Classification
CHAPTER 11 SERIAL ARRAY UNIT (continuation)		
p.407	Change of Transfer data length in **11.6.3 LIN transmission**	(a)
p.410	Change of Transfer data length in **11.6.4 LIN reception**	(a)
p.422	Change of **Figure 11-89. Initial Setting Procedure for Address Field Transmission**	(a)
p.423	Change of **Figure 11-90. Timing Chart of Address Field Transmission**	(a)
p.424	Change of **Figure 11-91. Flowchart of Address Field Transmission**	(a)
p.426	Change of **Figure 11-92. Example of Contents of Registers for Data Transmission of Simplified I^2C (IIC10)** and addition of **Note**	(a)
p.427	Change of **Figure 11-94. Flowchart of Data Transmission**	(a)
p.429	Change of **Figure 11-95. Example of Contents of Registers for Data Reception of Simplified I^2C (IIC10)** and addition of **Note**	(a)
p.430	Change of **Figure 11-96. Timing Chart of Data Reception**	(c)
p.430	Change of **Figure 11-97. Flowchart of Data Reception** and addition of **Caution**	(c)
p.431	Change of **Figure 11-99. Flowchart of Stop Condition Generation**	(c)
pp.437 to 441	Addition of **11.9 Relationship Between Register Settings and Pins**	(c)
CHAPTER 15 INTERRUPT FUNCTIONS		
p.538	Change of **Table 15-1. Interrupt Source List**	(a)
CHAPTER 17 STANDBY FUNCTION		
p.566	Addition of **Note** to **Figure 17-3. HALT Mode Release by Interrupt Request Generation**	(b)
p.570	Addition of **Note** to **Figure 17-5. Operation Timing When STOP Mode Is Released (When Unmasked Interrupt Request Is Generated)**	(b)
pp.571, 572	Addition of **Note** to **Figure 17-6. STOP Mode Release by Interrupt Request Generation**	(b)
CHAPTER 18 RESET FUNCTION		
p.574	Change of description in **(4)**	(c)
p.576	Change of **Figure 18-2. Timing of Reset by $\overline{\text{RESET}}$ Input**	(b)
p.577	Change of **Figure 18-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input**	(b)
CHAPTER 23 FLASH MEMORY		
p.612	Change of Pin No. in **Table 23-1. Wiring Between 78K0R/KE3 and Dedicated Flash Memory Programmer and addition** of **Note**	(b)
p.621	Change of **23.4.1 FLMD0 pin**	(c)
p.629	Change of **Remark** in **23.8 Flash Memory Programming by Self-Programming**	(e)
p.630	Change of **Figure 23-10. Flow of Self Programming (Rewriting Flash Memory)**, and addition of **Remark**	(b, e)
CHAPTER 25 BCD CORRECTION CIRCUIT		
pp.638, 639	Change of **25.3 BCD Correction Circuit Operation**	(b)
CHAPTER 26 INSTRUCTION SET		
p.641	Addition of addr5 to **Table 26-2. Symbols in "Operation" Column**	(c)
p.657	Change of operation of CALLT in **Table 26-5. Operation List (15/17)**	(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

(4/4)

Page	Description	Classification
CHAPTER 27 ELECTRICAL SPECIFICATIONS		
Throughout	Change of specifications of μPD78F1142, 78F1143, 78F1144, 78F1145, and 78F1146 from target specifications to formal specifications	(b)
p.660	**Absolute Maximum Ratings** • Change of Input voltage • Change of condition of Output voltage	(b)
p.663	Change of **Notes 1** in **Internal Oscillator Characteristics**	(c)
pp.665 to 669, 671, 672	**DC Characteristics** • Change of condition of Output current, high (I_{OH2}) • Change of condition of Output current, low (I_{OL2}) • Change of condition of Input voltage, high (V_{IH4}) • Change of condition of Input voltage, low (V_{IL4}) • Change of **Cautions 2** • Change of Output voltage, high (V_{OH2}) • Change of Output voltage, low (V_{OL2}) • Change of condition of Input leakage current, high (I_{LIH2}) • Change of condition of Input leakage current, low (I_{LIL2}) • Change of Supply current (I_{DD1}) and addition of low consumption current mode, **Notes 4**, and Remarks 4. • Change of Supply current (I_{DD2}) and addition of low consumption current mode, **Notes 4**, and Remarks 3.	(b)
pp.676 to 680	**AC Characteristics** **(1) Basic operation** • Addition of figures of **Minimum instruction execution time during main system clock operation** and **Minimum instruction execution time during self programming mode** in **(1) Basic operation** • Change of title in **AC Timing Test Points**	(b)
p.701	Change of figures and figure title in **Supply Voltage Rise Time Timing**	(c)
CHAPTER 28 PACKAGE DRAWINGS		
pp.706, 707	Addition of package drawing	(d)
APPENDIX A DEVELOPMENT TOOLS		
p.712	Change of **A.4.1 When using flash memory programmer FG-FP4 and FL-PR4**	(b)
p.713	Change of **A.4.2 When using on-chip debug emulator with programming function QB-MINI2**	(c)
pp.713, 714	Change of **A.5.1 When using in-circuit emulator QB-78K0RKX3**	(d)
p.714	Change of **A.5.2 When using on-chip debug emulator with programming function QB-MINI2**	(c)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

B.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/6)

Edition	Description	Chapter
4th edition	Change of status indication of μPD78F1142 and μPD78F1143 to "under development"	CHAPTER 1 OUTLINE
	1.1 Feature • Addition of single-power supply flash memory security function • Addition of flash shield window function to self-programming function	
	Changes of **Figure 3-1 Memory Map (μPD78F1142)** through **Figure 3-5 Memory Map (μPD78F1146)**	CHAPTER 3 CPU ARCHITECTURE
	Addition of **3.1.1(4) On-chip debug security ID setting area**	
	Addition of **Caution** to **3.1.3 Internal data memory space**	
	Addition of **Caution** to **3.2.4 Special function registers (SFRs)**	
	Change of **Note 1** in **Table 3-5 SFR List**	
	Change of BCD adjust result register in **Table 3-5 SFR List**	
	Addition of **Caution** to **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**	
	Addition of **Caution** to **Figure 5-7 Format of Peripheral Enable Register 0 (PER0)**	CHAPTER 5 CLOCK GENERATOR
	Addition of **Note 4** to **5.3 (7) Operation speed mode control register (OSMC)**	
	Change of description of **5.3 (8) Internal high-speed oscillator trimming register (HIOTRM)**	
	Addition of time until CPU operation start in **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))**	
	Change of **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))**	
	Addition of **Caution** to **5.6.1 (3) <3>**	
	Addition of **Caution 2** to **6.3 (1) Peripheral enable register 0 (PER0)**	CHAPTER 6 CLOCK GENERATOR
	Change of **Figure 6-6 Format of Timer Mode Register 0n (TMR0n)**	
	Addition of description to **6.3 (4) Timer status register 0n (TSR0n)**	
	Addition of **Table 6-3 OVF Bit Operation and Set/Clear Conditions in Each Operation Mode**	
	Addition of **Table 6-4 Operations from Count Operation Enabled State to TCR0n Count Start**, and (a) through (e)	
	Addition of description to **6.3 (11) Timer output level register 0 (TOL0)**	
	Change of description of **6.3 (12) Timer output mode register 0 (TOM0)**	
	Change of **Figure 6-20 Format of Timer Output Mode Register 0 (TOM0)** and **Remark**	
	Change of description to **Figure 6-21 Format of Input Switch Control Register (ISC)**	
	Addition of **6.4 Channel Output (TO0n pin) Control**	
	Addition of **6.5 Channel Input (TI0n Pin) Control**	

(2/6)

Edition	Description	Chapter
4th edition	Addition of MD0n0 bit condition to titles in the following figures • **Figure 6-37 Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MD0n0 = 1)** • **Figure 6-45 Example of Basic Timing of Operation as Frequency Divider (MD0n0 = 1)** • **Figure 6-49 Example of Block Diagram of Operation as Input Pulse Interval Measurement (MD0n0 = 0)**	CHAPTER 6 CLOCK GENERATOR
	Change of description of **6.7.3 Operation as frequency divider**	
	Change of description of **6.8.3 Operation as multiple PWM output function**	
	Change of clear conditions of real-time counter	CHAPTER 7 REAL-TIME COUNTER
	Change of description and **Caution 1** in **Figure 7-2 Format of Peripheral Enable Register 0 (PER0)**	
	Addition of **Caution 2** to **Figure 7-2 Format of Peripheral Enable Register 0 (PER0)**	
	Addition of **Caution** to **Figure 7-4 Format of Real-Time Counter Control Register 1 (RTCC1)**	
	Addition of **Caution** to **Figure 7-5 Format of Real-Time Counter Control Register 2 (RTCC2)**	
	Change of **Note 2** in **7.3 (5) Sub-count register (RSUBC)**	
	Change of description of **7.3 (8) Hour count register (HOUR)**	
	Change of bit name in **Figure 7-17 Format of Alarm Week Register (ALARMWW)**	
	Addition of **Caution 2** to **10.3 (1) Peripheral enable register 0 (PER0)**	CHAPTER 10 A/D CONVERTER
	Change of **Table 10-2 A/D Conversion Time Selection**	
	Addition of **Caution 3** to **11.3 (1) Peripheral enable register 0 (PER0)**	CHAPTER 11 SERIAL ARRAY UNIT
	Change of **Figure 11-7 Format of Serial Communication Operation Setting Register mn (SCRmn)**	
	Addition of description to **11.3 (13) Serial output level register m (SOLm)**	
	Changes of bits 1 and 3 in **Figure 11-16 Format of Serial Output Level Register m (SOLm)**	
	Changes of setting of **(a) Serial output register m (SOm)** and **Note** in **Figure 11-66 Example of Contents of Registers for UART Transmission of UART (UART0, UART1, UART2, UART3)**	
	Change of **Figure 11-89 Flowchart of Address Field Transmission**	
	Change of **Figure 11-92 Flowchart of Data Transmission**	
	Addition of **Caution 2** to **12.3 (1) Peripheral enable register 0 (PER0)**	CHAPTER 12 SERIAL INTERFACE IIC0
	Change of description of **12.5.4 (2) Selection clock setting method on the slave side**	
	Addition of description to **<1>** and **<3>** in **14.4.1 Operation procedure**	CHAPTER 14 DMA CONTROLLER
	Addition of description to **14.5.5 Forced termination by software**	
	Additions of description and **Note** to **14.6 (1) Priority of DMA**	
	Additions of reset processing time and clock supply stop time to the following figures • **Figure 17-4 HALT Mode Release by Reset** • **Figure 17-6 STOP Mode Release by Interrupt Request Generation** • **Figure 17-7 STOP Mode Release by Reset**	CHAPTER 17 STANDBY FUNCTION
	Change of **Figure 17-5 Operation Timing When STOP Mode Is Released (When Unmasked Interrupt Request Is Generated)**	

(3/6)

Edition	Description	Chapter
4th edition	Change of **Figure 18-2 Timing of Reset by $\overline{\text{RESET}}$ Input**	CHAPTER 18 RESET FUNCTION
	Change of **Figure 18-3 Timing of Reset Due to Watchdog Timer Overflow**	
	Change of **Figure 18-4 Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input**	
	Addition of reset processing time to **Figure 19-2 Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector**	CHAPTER 19 POWER-ON-CLEAR CIRCUIT
	Addition of **19.4 Caution for Power-on-Clear Circuit**	
	Addition of operation stabilization time	CHAPTER 20 LOW-VOLTAGE DETECTOR
	Change of **Caution 2** in **Figure 20-3 Format of Low-Voltage Detection Level Select Register (LVIS)**	
	Addition of **20.5 Caution for Low Voltage Detector**	
	Change of description of **22.1.1 (2) 000C1H/010C1H**	CHAPTER 22 OPTION BYTE
	Change of **Figure 22-2 Format of User Option Byte(000C1H/010C1H)**	
	Change of **Figure 22-4 Format of On-chip Debug Option Byte(000C3H/010C3H)**	
	Addition of description to **23. 4.1 (3) During writing by self programming**	CHAPTER 23 FLASH MEMORY
	Addition of description to **23.5 (1) Background event control register (BECTL)**	
	Addition of **23.6 Programming Method**	
	Addition of **23.7 Security Settings**	
	Addition of **23.8 Flash Memory Programming by Self-programming**	
	Addition of chapter	CHAPTER 24 ON-CHIP DEBUGGING
	Deletion of description of BCD correction carry register (BCDCY bit), etc.	CHAPTER 25 BCD CORRECTION CIRCUIT
	Absolute Maximum Ratings • Addition of regulator voltage (REGC) • Change of Input voltage and output voltage	CHAPTER 27 ELECTRICAL SPECIFICATIONS (TARGET)
	Addition of MIN. value and MAX. value in XT1 Oscillator Characteristics	
	DC characteristics • Change of **Note 1** in Output current, high (I_{OH1}) • Change of **Note 2** in Output current, low (I_{OL1}) • Addition of Supply current • Addition of Watchdog Timer operating current (I_{WDT}) • Addition of A/D Converter operating current (I_{ADC}) • Addition of DMA Controller operating current (I_{DMA}) • Addition of LVI operating current (I_{LVI})	
	Change of MIN. value of Conversion time (t_{CONV}) of **A/D Converter Characteristics**	
	Addition of **POC Circuit Characteristics**	
	Addition of **Supply Voltage Rise Time**	
	Addition of **LVI Circuit Characteristics**	
	Addition of **Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics**	
	Revision of chapter	APPENDIX A DEVELOPMENT TOOLS

(4/6)

Edition	Description	Chapter
4th edition (Modification Version)	Deletion of description of Temperature Correction function of Internal High-Speed Oscillation Clock and Temperature correction tables H, L from the following chapters. • CHAPTER 3 CPU ARCHITECTURE • CHAPTER 5 CLOCK GENERATOR • CHAPTER 10 A/D CONVERTER • CHAPTER 12 SERIAL INTERFACE IIC0 • CHAPTER 18 RESET FUNCTION • CHAPTER 27 ELECTRICAL SPECIFICATIONS (TARGET)	Throughout
5th edition	Deletion of target from the capacitance value of the capacitor connected to the REGC pin	Throughout
	Change of description in **2.2.15 REGC**	CHAPTER 2 PIN FUNCTIONS
	Modification of P60 to P64 in **Table 2-2 Connection of Unused Pins**	
	Addition (address change) of the BCDADJ register to **Table 3-6 Extended SFR (2nd SFR) List (1/4)**	CHAPTER 3 CPU ARCHITECTURE
	Change of **Figure 4-34 Bit Manipulation Instruction (P10)**	CHAPTER 4 PORT FUNCTIONS
	Change of **Caution 2** in **Figure 5-6 Format of System Clock Control Register (CKC)**	CHAPTER 5 CLOCK GENERATOR
	Change of description in **5.3 (8) Internal high-speed oscillator trimming register (HIOTRM)** and addition of **Caution**	
	Change of **Figure 5-9 Format of Internal High-Speed Oscillator Trimming Register (HIOTRM)** and addition of **Caution**	
	Change of **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))**	
	Addition of **Note** to **Figure 6-5 Format of Timer Clock Select Register 0 (TPS0)**	CHAPTER 6 TIMER ARRAY UNIT
	Change of **Table 6-3 OVF Bit Operation and Set/Clear Conditions in Each Operation Mode** and addition of **Remark**	
	Addition of **Caution 2** to **Figure 6-18. Format of Timer Output Register 0 (TO0)**	
	Change of description in **6.3 (14) Noise filter enable register 1 (NFEN1)**	
	Change of **6.5.1 TI0n edge detection circuit**	
	Change of **Figure 7-1 Block Diagram of Real-Time Counter**	CHAPTER 7 REAL-TIME COUNTER
	Addition of **Caution 3** to **Table 8-4 Setting Window Open Period of Watchdog Timer**	CHAPTER 8 WATCHDOG TIMER
	Fixing of the SOE01 and SOEm3 bit settings to “0”. Fixing of the SO10, SOm1, SOm3, CKO10, CKOm1, CKO12, and CKOm3 bit settings to “1”.	CHAPTER 11 SERIAL ARRAY UNIT
	Change of “Setting disabled (set to the initial value)” in **Remark**	
	Change of **Figure 11-1 Block Diagram of Serial Array Unit 0**	
	Change of **Figure 11-2 Block Diagram of Serial Array Unit 1**	
	Addition of settings and **Note** to **Figure 11-5 Format of Serial Clock Select Register m (SPSm)**	
	Change of **Figure 11-11 Format of Serial Channel Enable Status Register m (SEm)**	
	Change of **Figure 11-14 Format of Serial Output Enable Register m (SOEm)**	
	Addition of description to **11.3 (12) Serial output register m (SOm)**	

(5/6)

Edition	Description	Chapter
5th edition	Change of **Figure 11-15 Format of Serial Output Register m (SOm)**	CHAPTER 11 SERIAL ARRAY UNIT
	Addition of **Note** to transfer rate	
	Change of transfer rate and **Note** in **11.4.4 Slave transmission**	
	Change of transfer rate in **11.4.5 Slave reception**	
	Change of transfer rate in **11.4.6 Slave transmission/reception**	
	Change of **Note** in **11.4.7 (2)**	
	Addition of setting and **Note** to **Table 11-2 Operating Clock Selection**	
	Change of transfer rate and addition of **Note**	
	Change of **Figure 11-66 Example of Contents of Registers for UART Transmission of UART (UART0, UART1, UART3)**	
	Change of **Figure 11-74 Example of Contents of Registers for UART Reception of UART (UART0, UART1, UART3)**	
	Change of **Figure 11-77 Procedure for Resuming UART Reception**	
	Addition of setting and **Note** to **Table 11-3 Operating Clock Selection**	
	Change of **Figure 11-92 Flowchart of Data Transmission**	
	Addition of setting and **Note** to **Table 11-4 Operating Clock Selection**	
	Change of **Figure 14-9 Example of Setting for UART Consecutive Reception + ACK Transmission**	CHAPTER 14 DMA CONTROLLER
	Additions of description to **14.6 (4) DMA pending instruction**	
	Change of **Figure 17-4 HALT Mode Release by Reset**	CHAPTER 17 STANDBY FUNCTION
	Change of **Figure 17-7 STOP Mode Release by Reset**	
	Change of reset processing in **Figure 18-2 Timing of Reset by $\overline{\text{RESET}}$ Input**	CHAPTER 18 RESET FUNCTION
	Change of reset processing in **Figure 18-4 Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input**	
	Change of **Caution 2** in **Figure 18-5 Format of Reset Control Flag Register (RESF)**	
	Change of **Figure 19-2 Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)**	CHAPTER 19 POWER-ON-CLEAR CIRCUIT
	Change of **Figure 19-2 Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)** and addition of **Note**	
	Change of **Figure 19-3 Example of Software Processing After Reset Release**	
	Change of **Note 4** in **Figure 20-2 Format of Low-Voltage Detection Register (LVIM)** and addition of **Caution 3**	CHAPTER 20 LOW-VOLTAGE DETECTOR
	Change of **Caution 2** in **Figure 20-3 Format of Low-Voltage Detection Level Select Register (LVIS)**	
	Change of <5> in **20.4.1 (1) (a)**	
	Change of **Note 2** in **Figure 20-5 Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 1)**	
	Change of description and **Caution** in **20.4.1 (1) (b)**	
	Change of **Figure 20-6 Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 0)** and **Note**	
	Change of <4> in **20.4.1 (2)**	
	Change of **Figure 20-7 Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 1)** and **Note 2**	
	Change of <5> in **20.4.2 (1)**	

(6/6)

Edition	Description	Chapter
5th edition	Additions of **Note 3** to **Figure 20-8 Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 1)**	CHAPTER 20 LOW-VOLTAGE DETECTOR
	Change of description and **Caution** in **20.4.2 (1) (b)**	
	Change of **Figure 20-9 Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 0)** and addition of **Note**	
	Change of <4> in **20.4.2 (2)**	
	Change of **Figure 20-10 Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 1)** and addition of **Note 3**	
	Change of **Figure 20-11 Example of Software Processing After Reset Release**	
	Change of **21.1 Regulator Overview**	CHAPTER 21 REGULATOR
	Addition of **Note 3** to **Figure 21-1 Format of Regulator Mode Control Register (RMC)**	
	Change of description in **22.1.1 (2) 000C1H/010C1H**	CHAPTER 22 OPTION BYTE
	Change of **Figure 22-2 Format of User Option Byte (000C1H/010C1H)** and **Caution 2**	
	Change of description in **23.4.5 REGC pin**	CHAPTER 23 FLASH MEMORY
	Addition of **Caution 4** to **23.8 Flash Memory Programming by Self-Programming**	
	Addition of **24.3 Securing of user resources**	CHAPTER 24 ON-CHIP DEBUGGING
	Modification of throughout	CHAPTER 27 ELECTRICAL SPECIFICATIONS (TARGET)

For further information,
please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
http://www.cn.necel.com/

Shanghai Branch
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai, P.R.China P.C:200120
Tel:021-5888-5400
http://www.cn.necel.com/

Shenzhen Branch
Unit 01, 39/F, Excellence Times Square Building,
No. 4068 Yi Tian Road, Futian District, Shenzhen,
P.R.China P.C:518048
Tel:0755-8282-9800
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
Unit 1601-1613, 16/F., Tower 2, Grand Century Place,
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
http://www.tw.necel.com/

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
http://www.kr.necel.com/

G0706